MapServer with OSGeo4W Users Guide

Last updated: 2010-11-16

MapServer
open source web mapping

About the MapServer with OSGeo4W Users Guide

The “MapServer with OSGeo4W Users Guide” was developed through funding from the Japan Science and Technology Agency, through Japanese project partners Osaka City University, Tezukayamagakuin University, and Applied Technology Co., as well as Gateway Geomatics in Canada.

The users guide has been adapted from the “MapServer Users Guide” originally created in 2004 by Osaka City University and Orkney Inc. of Japan.

Table of Contents

Background Technical Information

This user guide refers to the use of MapServer (http://www.mapserver.org/), which is Open Source software that is used to publish and share your spatial information through the Internet. MapServer contains many excellent documents that you should be familiar with, including:

- Introduction to MapServer: http://www.mapserver.org/introduction.html
- Mapfile Reference: http://www.mapserver.org/mapfile/
- Data Input in MapServer: http://www.mapserver.org/input/
- OGC Support in MapServer: http://www.mapserver.org/ogc/

This user guide is based on MapServer version 5.6.5.

Installing MapServer with OSGeo4W

OSGeo4W (http://trac.osgeo.org/osgeo4w/) is an installer for the Windows operating system that contains many useful software packages related to the Open Source Geospatial Foundation (http://www.osgeo.org/).

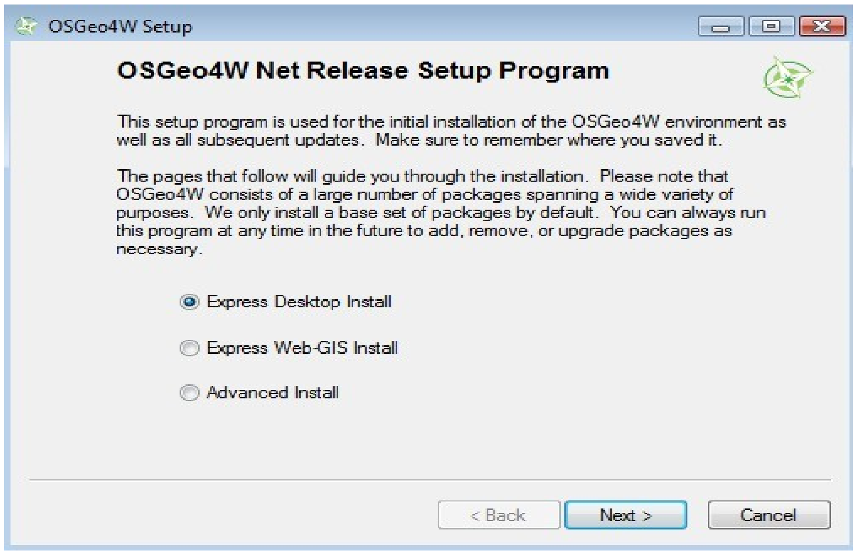
Step 1: Download OSGeo4W

Download the OSGeo4W installer from: http://download.osgeo.org/osgeo4w/osgeo4w-setup.exe

Save the file onto your desktop.

Step 2: Start the OSGeo4W Installer

Double-click the 'osgeo4w-setup.exe' file. You should see the following opening window of OSGeo4W:


OSGeo4W Setup

OSGeo4W Net Release Setup Program

This setup program is used for the initial installation of the OSGeo4W environment as well as all subsequent updates. Make sure to remember where you saved it.

The pages that follow will guide you through the installation. Please note that OSGeo4W consists of a large number of packages spanning a wide variety of purposes. We only install a base set of packages by default. You can always run this program at any time in the future to add, remove, or upgrade packages as necessary.

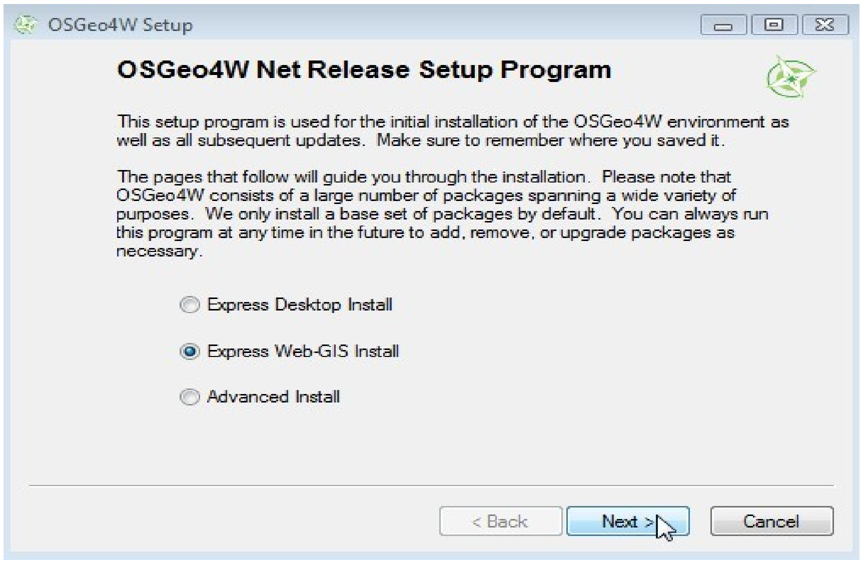
[✓ Express Desktop Install]

Express Web-GIS Install

Advanced Install

Step 3: Select 'Express Web-GIS Install' Option

Click on the 'Express Web-GIS Install' option, and click on the 'Next' button.



OSGeo4W Setup

OSGeo4W Net Release Setup Program

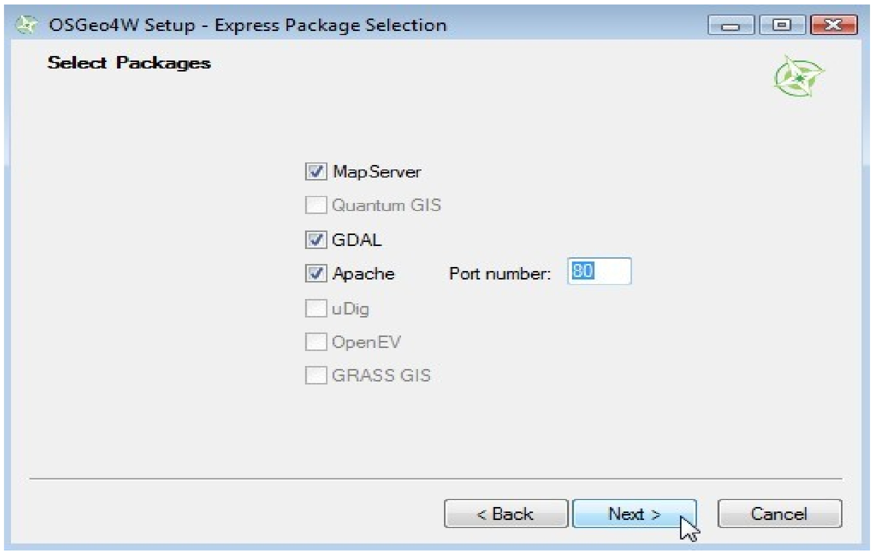
This setup program is used for the initial installation of the OSGeo4W environment as well as all subsequent updates. Make sure to remember where you saved it.

The pages that follow will guide you through the installation. Please note that OSGeo4W consists of a large number of packages spanning a wide variety of purposes. We only install a base set of packages by default. You can always run this program at any time in the future to add, remove, or upgrade packages as necessary.

- ○ Express Desktop Install
- ● Express Web-GIS Install
- ○ Advanced Install

Step 4: Select Packages to Install

Since we are installing MapServer and its dependent libraries, we can leave all of the packages selected (MapServer, GDAL, Apache) and click on the next button. You can optionally specify the Apache port number to use, although leaving the port as 80 is recommended. Click the “Next” button to begin installation.



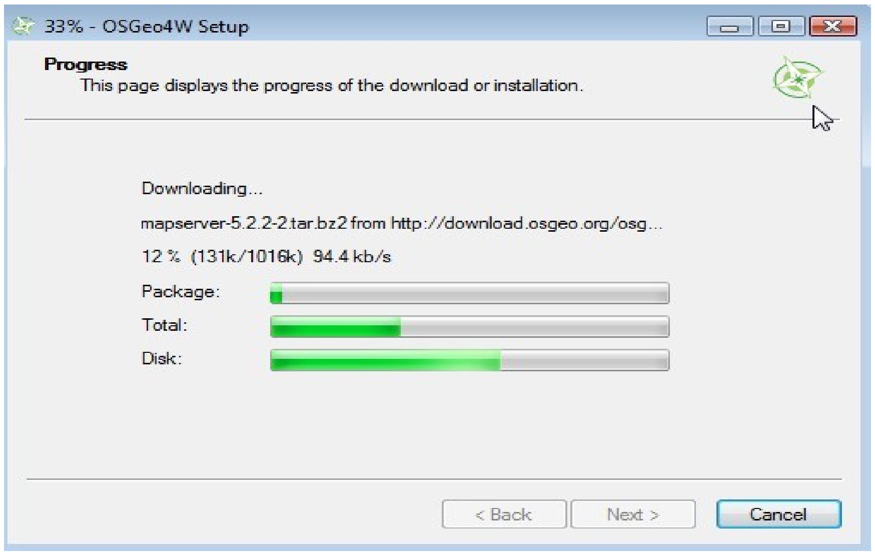
OSGeo4W Setup - Express Package Selection

Select Packages

- ☑ MapServer
- ☐ Quantum GIS
- ☑ GDAL
- ☑ Apache   Port number: 80
- ☐ uDig
- ☐ OpenEV
- ☐ GRASS GIS

Step 5: Download Packages

The installer should now automatically fetch and install MapServer and all of its libraries from the OSGeo download server.

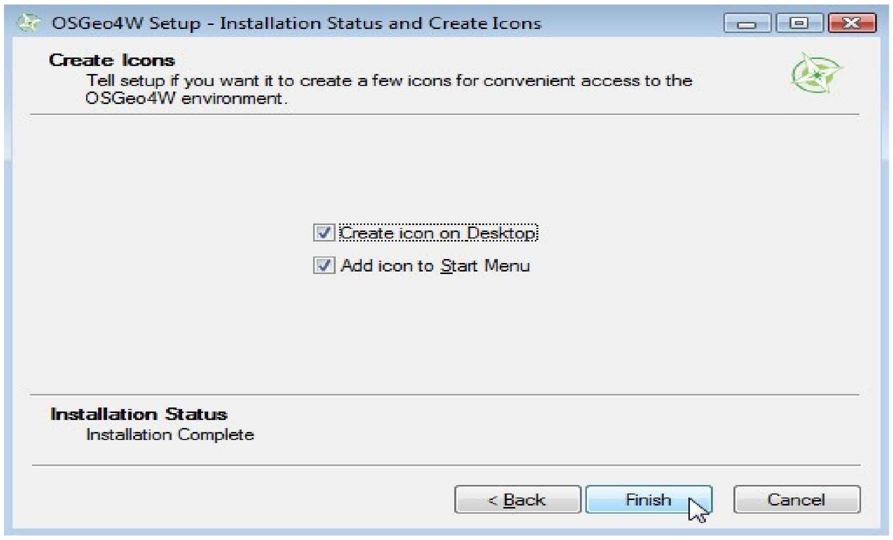


Progress
This page displays the progress of the download or installation.

Downloading...
mapserver-5.2.2-2.tar.bz2 from http://download.osgeo.org/osg...
12 % (131k/1016k) 94.4 kb/s
Package:
Total:
Disk:

Step 6: Finish Installation

Once downloading and installation from the OSGeo download server is complete, you can now optionally choose to create shortcuts for OSGeo4W on your desktop, and also in your Start Menu (both are recommended). Click the "Finish" button to finalize the installation.



Create Icons
Tell setup if you want it to create a few icons for convenient access to the OSGeo4W environment.

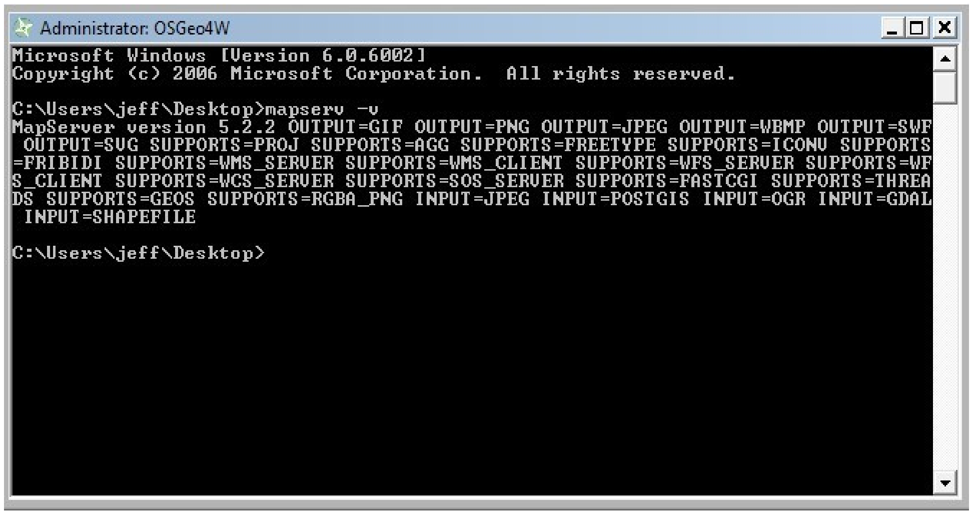
☑ Create icon on Desktop
☑ Add icon to Start Menu

Installation Status
Installation Complete

Step 7: Verify your MapServer Installation

You can verify that MapServer is installed through several ways.

1. click on the OSGeo4W shortcut on your desktop – a command window should be opened (*note: Vista and Windows7 users should first right-click on the OSGeo4W shortcut and select "Run as Administrator"*)
2. type the following in the command window: mapserv -v
3. the MapServer version should be returned, such as:

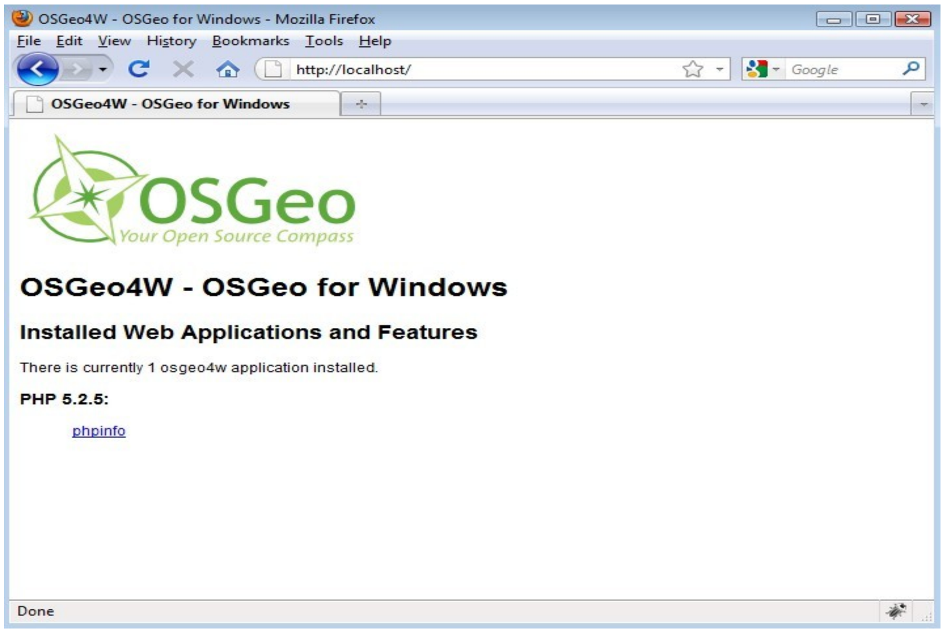


```
Microsoft Windows [Version 6.0.6002]
Copyright (c) 2006 Microsoft Corporation.  All rights reserved.

C:\Users\jeff\Desktop>mapserv -v
MapServer version 5.2.2 OUTPUT=GIF OUTPUT=PNG OUTPUT=JPEG OUTPUT=WBMP OUTPUT=SWF
 OUTPUT=SVG SUPPORTS=PROJ SUPPORTS=AGG SUPPORTS=FREETYPE SUPPORTS=ICONV SUPPORTS
=FRIBIDI SUPPORTS=WMS_SERVER SUPPORTS=WMS_CLIENT SUPPORTS=WFS_SERVER SUPPORTS=WF
S_CLIENT SUPPORTS=WCS_SERVER SUPPORTS=SOS_SERVER SUPPORTS=FASTCGI SUPPORTS=THREA
DS SUPPORTS=GEOS SUPPORTS=RGBA_PNG INPUT=JPEG INPUT=POSTGIS INPUT=OGR INPUT=GDAL
 INPUT=SHAPEFILE

C:\Users\jeff\Desktop>
```

4. Verify that Apache was installed properly by opening a Web browser, such as Internet Explorer or Firefox, and going to the location http://localhost/ You should see an OSGeo4W page such as:


OSGeo4W - OSGeo for Windows - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/
Google
OSGeo4W - OSGeo for Windows
OSGeo
Your Open Source Compass
Done

OSGeo4W - OSGeo for Windows

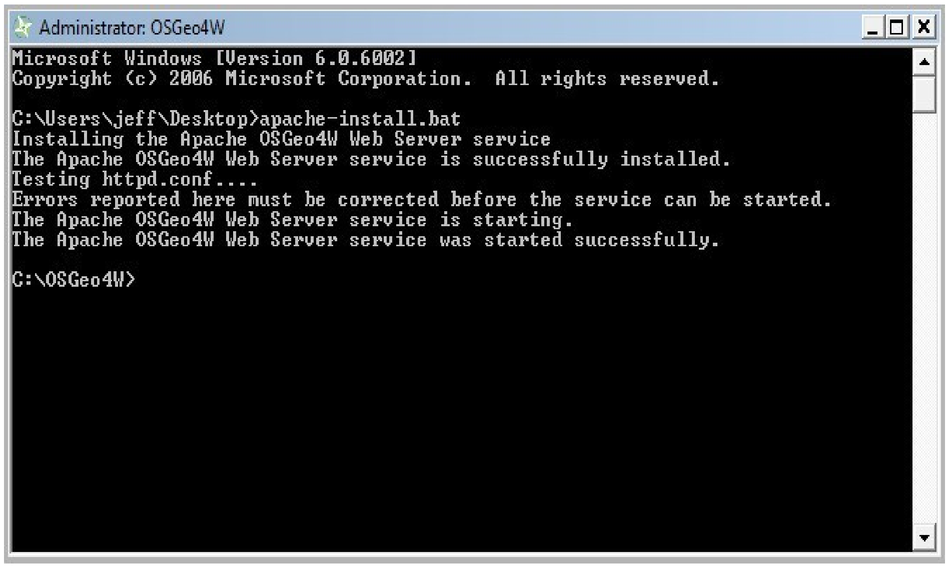
Installed Web Applications and Features

There is currently 1 osgeo4w application installed.

PHP 5.2.5:

phpinfo

5. If you do not see the OSGeo4W opening page in your Web browser:
   - right-click on the OSGeo4W desktop shortcut and select "Run as Administrator"
   - type the following in the command window: apache-install.bat
   - you should see a message that Apache was installed and started successfully, such as:



```
Microsoft Windows [Version 6.0.6002]
Copyright (c) 2006 Microsoft Corporation.  All rights reserved.

C:\Users\jeff\Desktop>apache-install.bat
Installing the Apache OSGeo4W Web Server service
The Apache OSGeo4W Web Server service is successfully installed.
Testing httpd.conf....
Errors reported here must be corrected before the service can be started.
The Apache OSGeo4W Web Server service is starting.
The Apache OSGeo4W Web Server service was started successfully.

C:\OSGeo4W>
```

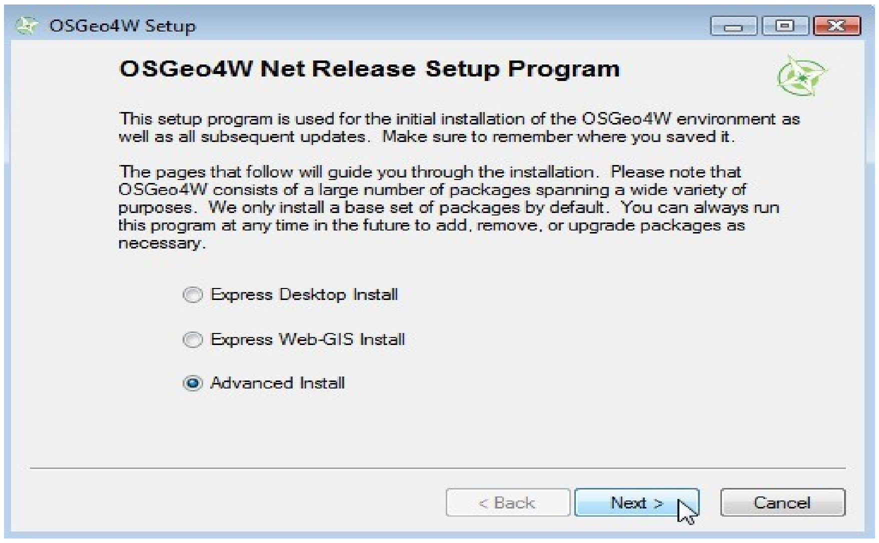
   - now re-try http://localhost/ in your Web browser

Using MapServer CGI to Create an Application

The following section is based on the "MapServer Tutorial" package for OSGeo4W. Execute the following to install the package, which includes data and all application files.

Install the "MapServer Tutorial" OSGeo4W Package

1) Start the 'osgeo4w-setup.exe' installer and select the "Advanced Install" option.



OSGeo4W Setup

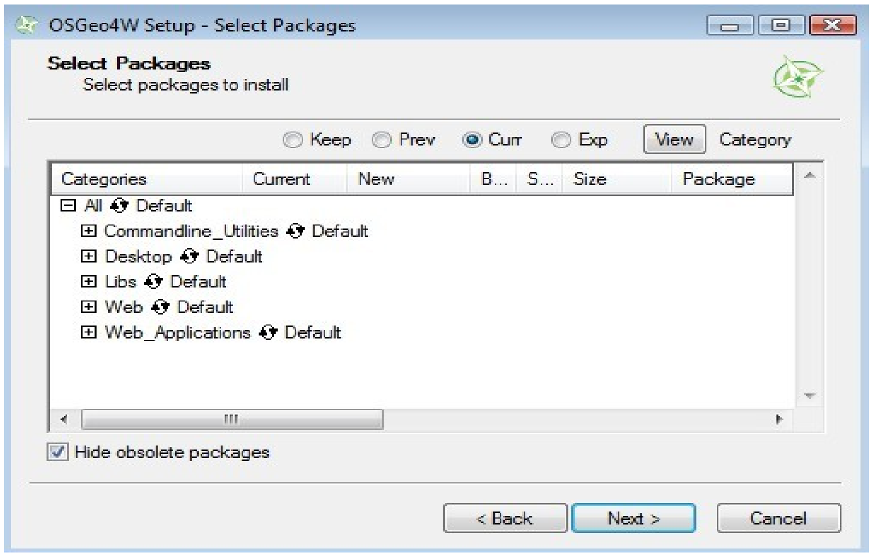
OSGeo4W Net Release Setup Program

This setup program is used for the initial installation of the OSGeo4W environment as well as all subsequent updates. Make sure to remember where you saved it.

The pages that follow will guide you through the installation. Please note that OSGeo4W consists of a large number of packages spanning a wide variety of purposes. We only install a base set of packages by default. You can always run this program at any time in the future to add, remove, or upgrade packages as necessary.

○ Express Desktop Install
○ Express Web-GIS Install
[✓ Advanced Install]

2) Advance to the "Select Packages" window.



OSGeo4W Setup - Select Packages

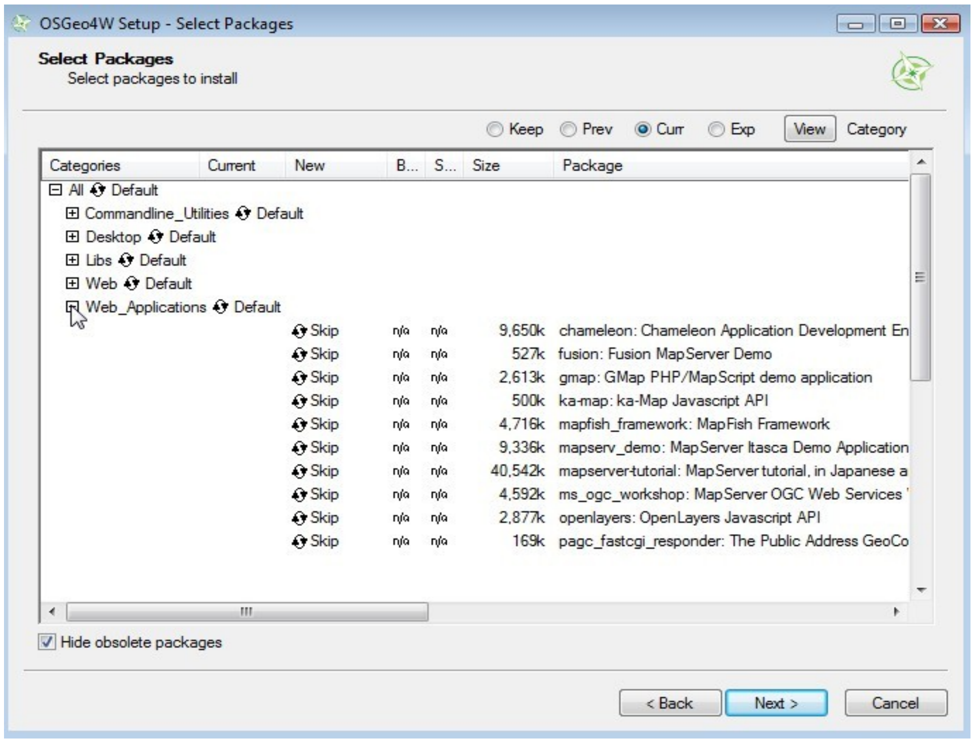
Select Packages
Select packages to install

○ Keep ○ Prev [✓ Curr] ○ Exp View Category

Categories	Current	New	B...	S...	Size	Package
⊟ All ↻ Default						
⊞ Commandline_Utilities ↻ Default						
⊞ Desktop ↻ Default						
⊞ Libs ↻ Default						
⊞ Web ↻ Default						
⊞ Web_Applications ↻ Default						

☑ Hide obsolete packages

3) Expand the "Web_Applications" tree.



OSGeo4W Setup - Select Packages

Select Packages
Select packages to install

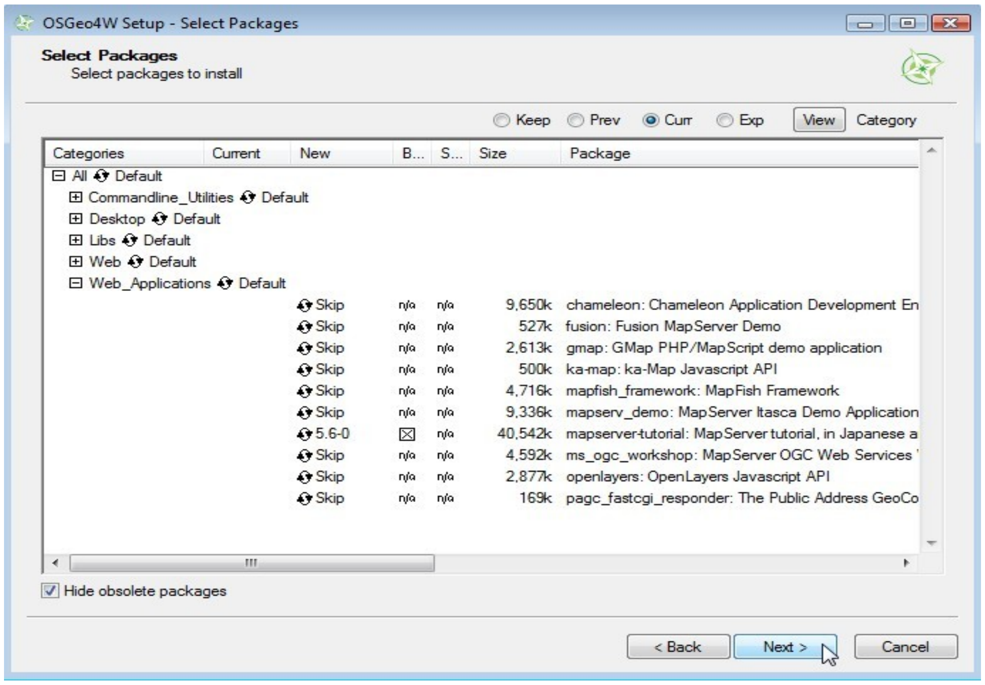
○ Keep ○ Prev ◉ Curr ○ Exp   View   Category

Categories	Current	New	B...	S...	Size	Package
⊟ All Default						
⊞ Commandline_Utilities Default						
⊞ Desktop Default						
⊞ Libs Default						
⊞ Web Default						
⊟ Web_Applications Default						
		Skip	n/a	n/a	9,650k	chameleon: Chameleon Application Development En
		Skip	n/a	n/a	527k	fusion: Fusion MapServer Demo
		Skip	n/a	n/a	2,613k	gmap: GMap PHP/MapScript demo application
		Skip	n/a	n/a	500k	ka-map: ka-Map Javascript API
		Skip	n/a	n/a	4,716k	mapfish_framework: MapFish Framework
		Skip	n/a	n/a	9,336k	mapserv_demo: MapServer Itasca Demo Application
		Skip	n/a	n/a	40,542k	mapserver-tutorial: MapServer tutorial, in Japanese a
		Skip	n/a	n/a	4,592k	ms_ogc_workshop: MapServer OGC Web Services
		Skip	n/a	n/a	2,877k	openlayers: OpenLayers Javascript API
		Skip	n/a	n/a	169k	pagc_fastcgi_responder: The Public Address GeoCo

☑ Hide obsolete packages

< Back   Next >   Cancel

4) Install the "mapserver-tutorial" package. Cick on the "skip" beside the "mapserver-tutorial" text, so that a version number is listed, and then click <Next>.



OSGeo4W Setup - Select Packages

Select Packages
Select packages to install

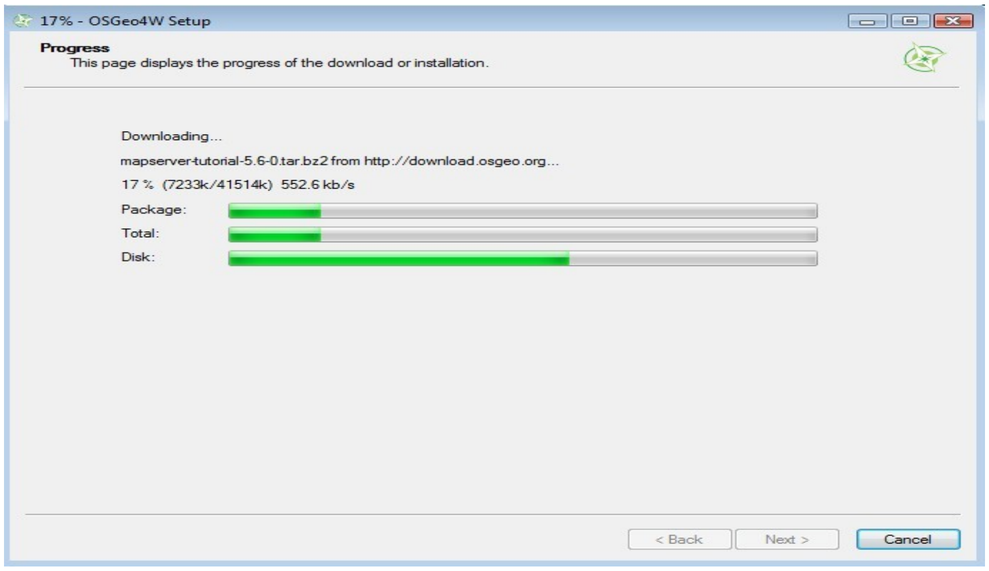
○ Keep ○ Prev ◉ Curr ○ Exp   View   Category

Categories	Current	New	B...	S...	Size	Package
⊟ All Default						
⊞ Commandline_Utilities Default						
⊞ Desktop Default						
⊞ Libs Default						
⊞ Web Default						
⊟ Web_Applications Default						
		Skip	n/a	n/a	9,650k	chameleon: Chameleon Application Development En
		Skip	n/a	n/a	527k	fusion: Fusion MapServer Demo
		Skip	n/a	n/a	2,613k	gmap: GMap PHP/MapScript demo application
		Skip	n/a	n/a	500k	ka-map: ka-Map Javascript API
		Skip	n/a	n/a	4,716k	mapfish_framework: MapFish Framework
		Skip	n/a	n/a	9,336k	mapserv_demo: MapServer Itasca Demo Application
		5.6-0	☒	n/a	40,542k	mapserver-tutorial: MapServer tutorial, in Japanese a
		Skip	n/a	n/a	4,592k	ms_ogc_workshop: MapServer OGC Web Services
		Skip	n/a	n/a	2,877k	openlayers: OpenLayers Javascript API
		Skip	n/a	n/a	169k	pagc_fastcgi_responder: The Public Address GeoCo

☑ Hide obsolete packages

< Back   Next >   Cancel

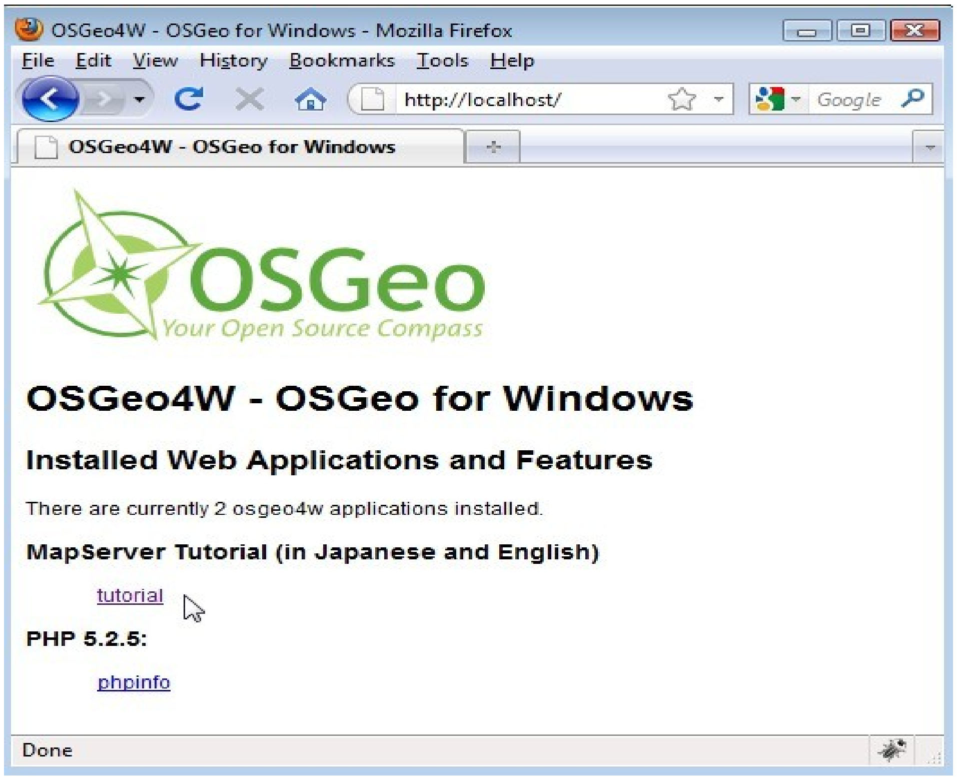
5) The tutorial files will automatically download from the OSGeo download server.



Progress
This page displays the progress of the download or installation.

Downloading...
mapserver-tutorial-5.6-0.tar.bz2 from http://download.osgeo.org...
17 % (7233k/41514k) 552.6 kb/s
Package:
Total:
Disk:

6) Open your Web browser (such as Internet Explorer or Firefox) and goto http://localhost/ - you should now see a section on the page containing the MapServer Tutorial.


OSGeo4W - OSGeo for Windows - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/
Google
OSGeo4W - OSGeo for Windows
OSGeo
Your Open Source Compass
Done

OSGeo4W - OSGeo for Windows

Installed Web Applications and Features

There are currently 2 osgeo4w applications installed.

MapServer Tutorial (in Japanese and English)

tutorial

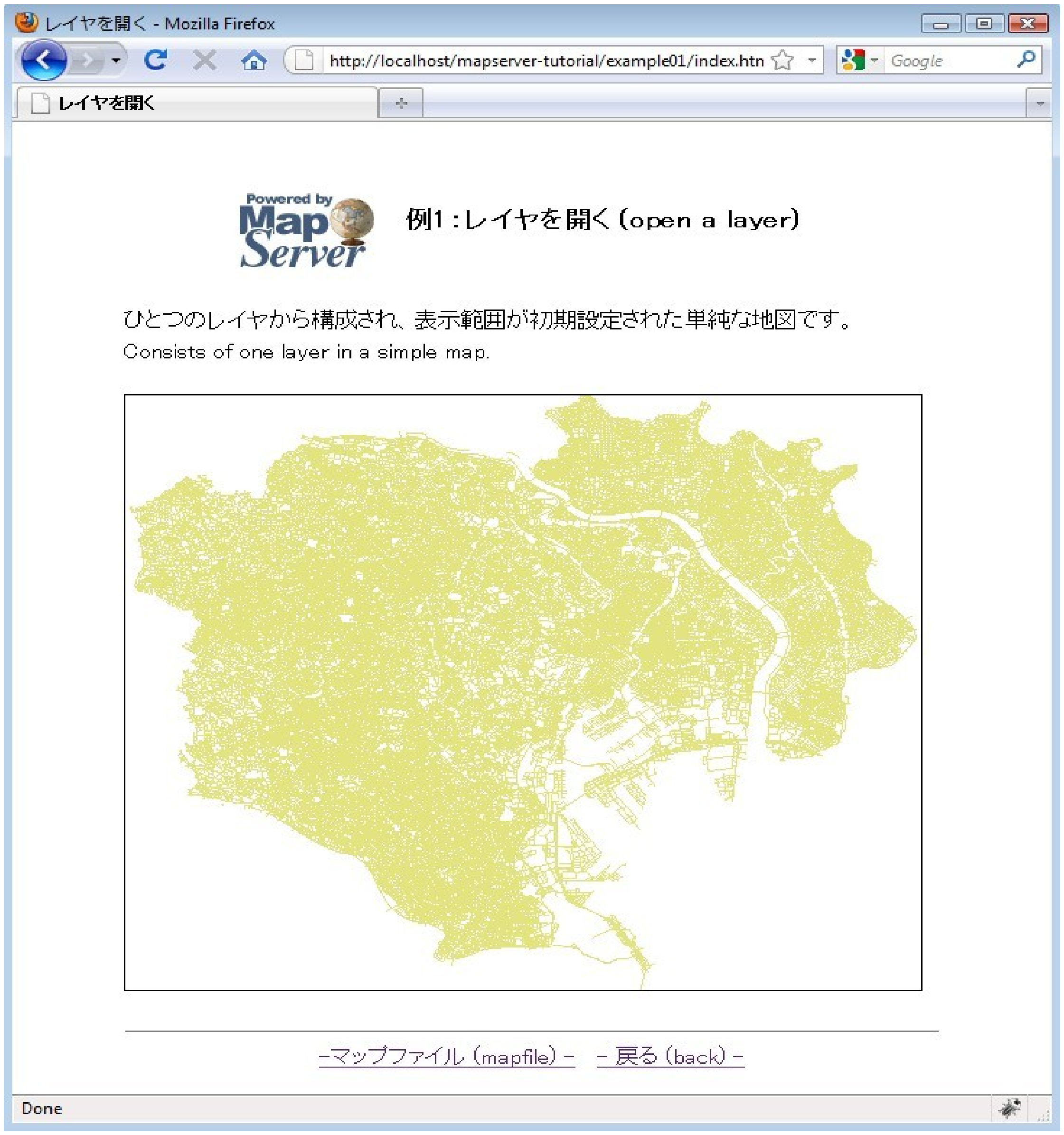
PHP 5.2.5:

phpinfo

7) Goto the MapServer tutorial by clicking on the "tutorial" link. The opening page for the tutorial should be displayed.

Exercise 1: Display a Layer

This example consists of a single map layer. MapServer stores data configuration parameters in a *.map* file. More information about mapfiles can be found at: http://www.mapserver.org/mapfile/


レイヤを開く - Mozilla Firefox
http://localhost/mapserver-tutorial/example01/index.htm
Google
レイヤを開く
Powered by MapServer

例1：レイヤを開く (open a layer)

ひとつのレイヤから構成され、表示範囲が初期設定された単純な地図です。
Consists of one layer in a simple map.

-マップファイル (mapfile) - - 戻る (back) -
Done

The mapfile used in this exercise is:

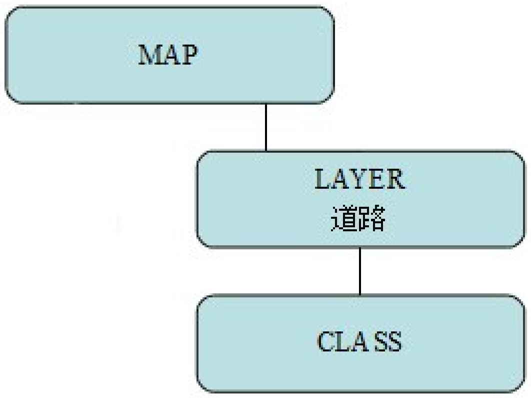
```
MAP
  IMAGETYPE    PNG
  EXTENT       139.562947 35.524705 139.917821 35.817635
  SIZE         550 450

  IMAGECOLOR   255 255 255
  SHAPEPATH    "../data"

  # Start of LAYER DEFINITIONS-------------------------------
  LAYER
    NAME "‶¹~H"
    DATA dourokukan
    STATUS DEFAULT
    TYPE LINE

    CLASS
      NAME "‶¹~H"
      STYLE
        COLOR 227 227 127
      END
    END
  END
  # End of LAYER DEFINITIONS -------------------------------
END
```

The MapServer mapfile is based on objects. Each object contains parameters or other objects. All valid objects and parameters are documented at: http://www.mapserver.org/mapfile/. The mapfile has a hierarchical structure; the MAP object is always the top-level object, and all other objects exist within the MAP object (see the following image). Each object has an "END" parameter, that signals the closing of the object. Comments in a mapfile are lines that begin with the "#" character.



MAP
This is the top-level object, and is required for every mapfile.

IMAGETYPE
The imagetype parameter is used to specify the output format of the map image.

EXTENT
The extent parameter specifies the output range of the map. A useful tool to retrieve extents of a file is the commandline *ogrinfo.exe* tool, which you can use through the OSGeo4W shell shortcut on your desktop.

SIZE
The size values are the size in pixels of the output map image.

IMAGECOLOR
This is the color of the background of the map, in RGB format.

SHAPEPATH
This is the path to the folder containing the spatial data. You can use absolute paths (full paths) to the data directory, or relative paths to the data directory (relative to the mapfile).

LAYER
This marks the start of a LAYER object. Since MapServer version 5.0, there is no limit to the number of layers you can have in a mapfile.

NAME
A string identifier for the layer.

DATA
The data filename, relative to the SHAPEPATH. If you do not specify an extension, MapServer will assume "*.shp*". More information on the many types of data that MapServer can read directly can be found at: http://www.mapserver.org/input/

STATUS
Tells MapServer to show the layer or not. Valid values are ON/OFF/DEFAULT. DEFAULT is required for MapServer CGI (if you are calling *mapserv.exe*).

TYPE
The geometry type of the data. Common types are POINT/LINE/POLYGON for vector data, and also RASTER for raster data. Consult the mapfile reference for all valid types.

CLASS
This marks the start of a CLASS object. CLASS objects are used with STYLE objects to classify the data.

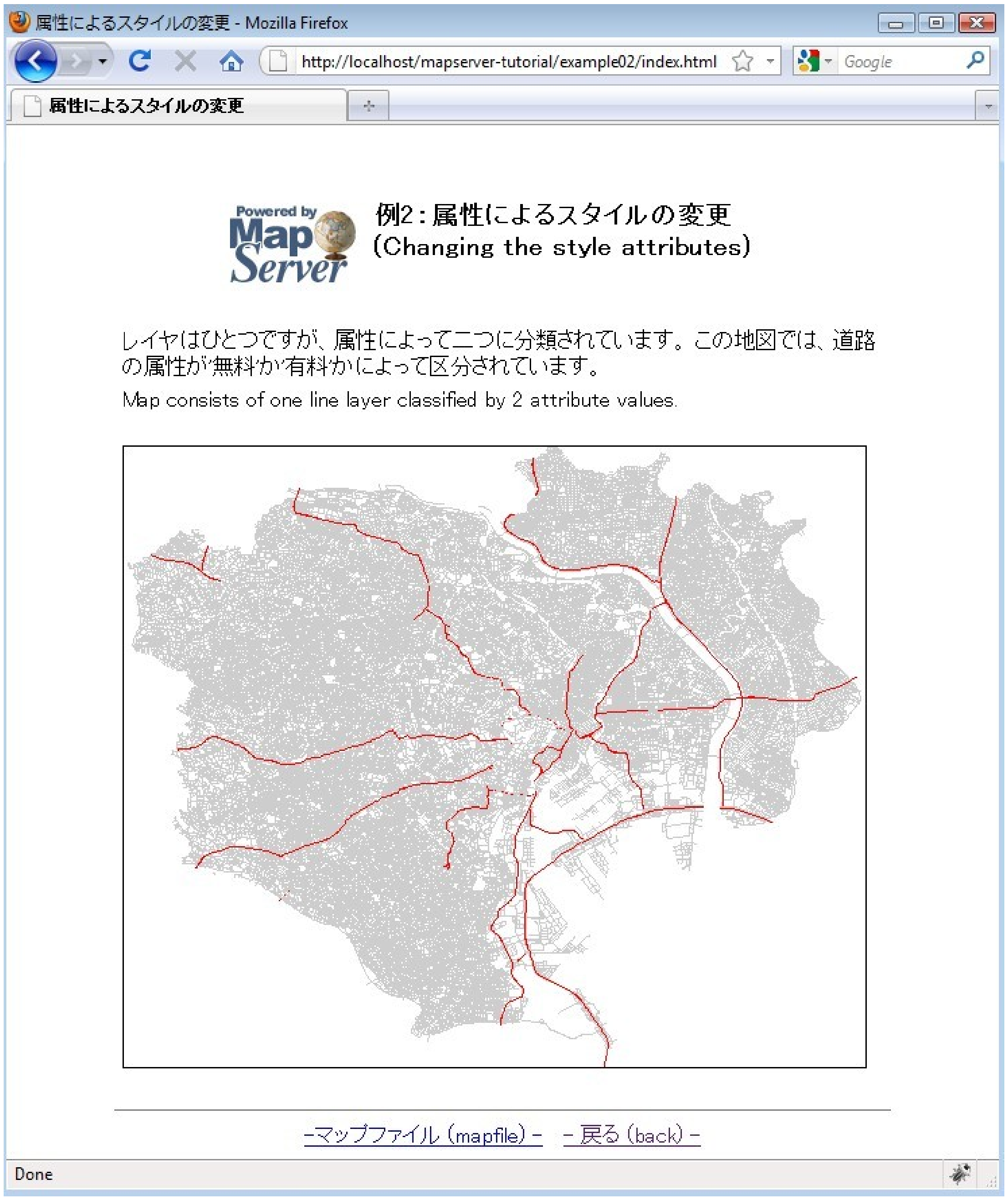
NAME
It is good practice to always provide a name for each your CLASS objects.

STYLE
This marks the start of a STYLE object. The STYLE object contains colors needed for your layer. A single CLASS object can have several STYLE objects.

COLOR
This is a color in RGB format.

Exercise 2: Change Style Attributes

This example consists of a single map layer classified by two attribute values.



属性によるスタイルの変更 - Mozilla Firefox

http://localhost/mapserver-tutorial/example02/index.html

例2：属性によるスタイルの変更
(Changing the style attributes)

レイヤはひとつですが、属性によって二つに分類されています。この地図では、道路の属性が無料か有料かによって区分されています。

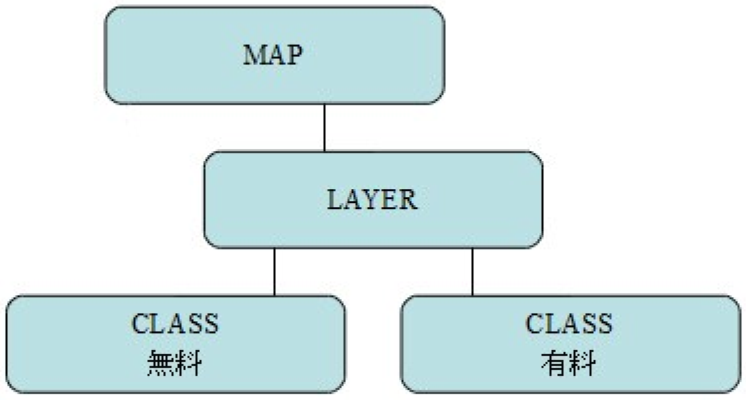
Map consists of one line layer classified by 2 attribute values.

-マップファイル (mapfile)- -戻る (back)-

Done

The mapfile used in this exercise is:

```
MAP
  IMAGETYPE   PNG
  EXTENT      139.562947 35.524705 139.917821 35.817635
  SIZE        550 450
  IMAGECOLOR  255 255 255
  SHAPEPATH   "../data"
  LAYER
    NAME "“¹˜H"
    DATA dourokukan
    STATUS DEFAULT
    TYPE LINE
    CLASSITEM "YURYO"
    CLASS
      NAME "first class"
      EXPRESSION '–³—¿'
      STYLE
        COLOR 205 205 205
      END
    END
    CLASS
      NAME "second class"
      EXPRESSION '—L—¿'
      STYLE
        COLOR 255 0 0
      END
    END
  END
END
```



As you can see in the mapfile, the line layer now contains two CLASS objects, used to style the two attributes. This mapfile contains a couple of important parameters:

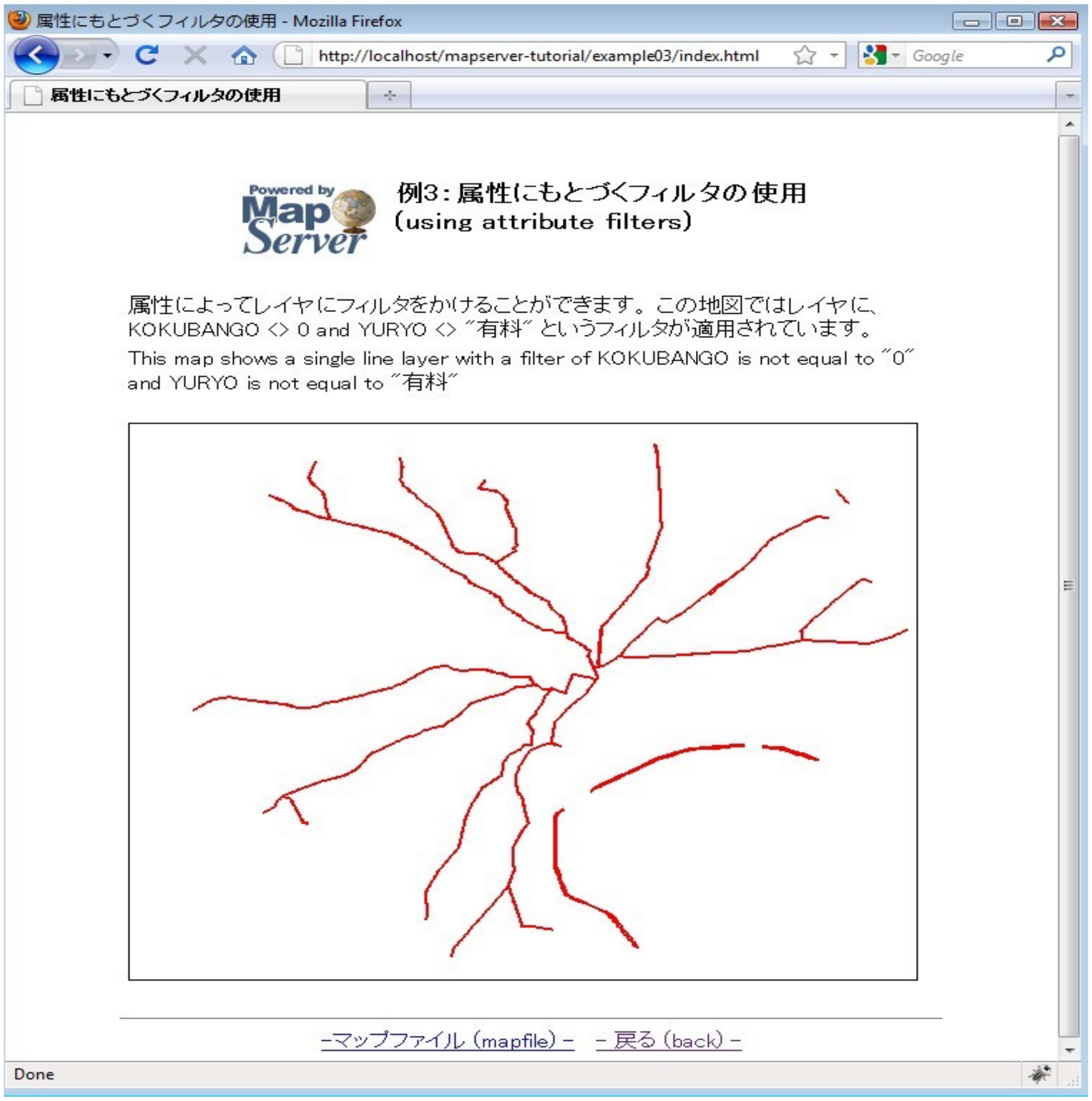
CLASSITEM

The attribute field used to classify the layer. In this example, the attribute "YURYO" is used; if you open *dourokukan.dbf* you will see that the field exists there.

EXPRESSION
The value to use for each class. More information about MapServer expressions can be found at: http://www.mapserver.org/mapfile/expressions.html

Exercise 3: Using Attribute Filters

This example consists of a single map layer using an attribute filter.


属性にもとづくフィルタの使用 - Mozilla Firefox
http://localhost/mapserver-tutorial/example03/index.html
Google
属性にもとづくフィルタの使用
Powered by MapServer
Done

例3：属性にもとづくフィルタの使用
(using attribute filters)

属性によってレイヤにフィルタをかけることができます。この地図ではレイヤに、KOKUBANGO <> 0 and YURYO <> "有料" というフィルタが適用されています。
This map shows a single line layer with a filter of KOKUBANGO is not equal to "0" and YURYO is not equal to "有料"

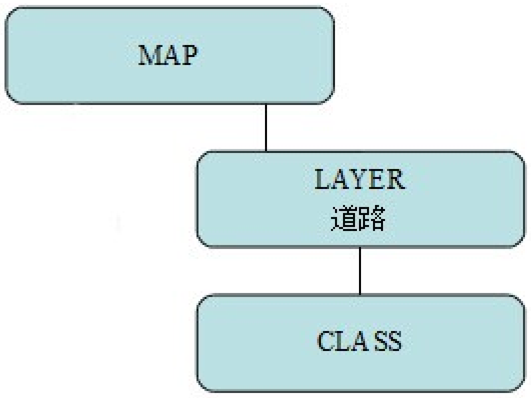
-マップファイル (mapfile) - - 戻る (back) -

The mapfile used in this exercise is:

```
MAP
 IMAGETYPE   PNG
 EXTENT      139.562947 35.524705 139.917821 35.817635
 SIZE        550 450
 IMAGECOLOR  255 255 255
 SHAPEPATH   "../data"

 LAYER
  NAME "“¹˜H"
  DATA dourokukan
  STATUS DEFAULT
  TYPE LINE
  FILTER  ("[KOKUBANGO]" ne "0" and "[YURYO]" ne "—L—¿" )
  CLASS
   NAME "“¹˜H"
   STYLE
    COLOR 250 0 0
    WIDTH 2
   END
  END
 END

END
```

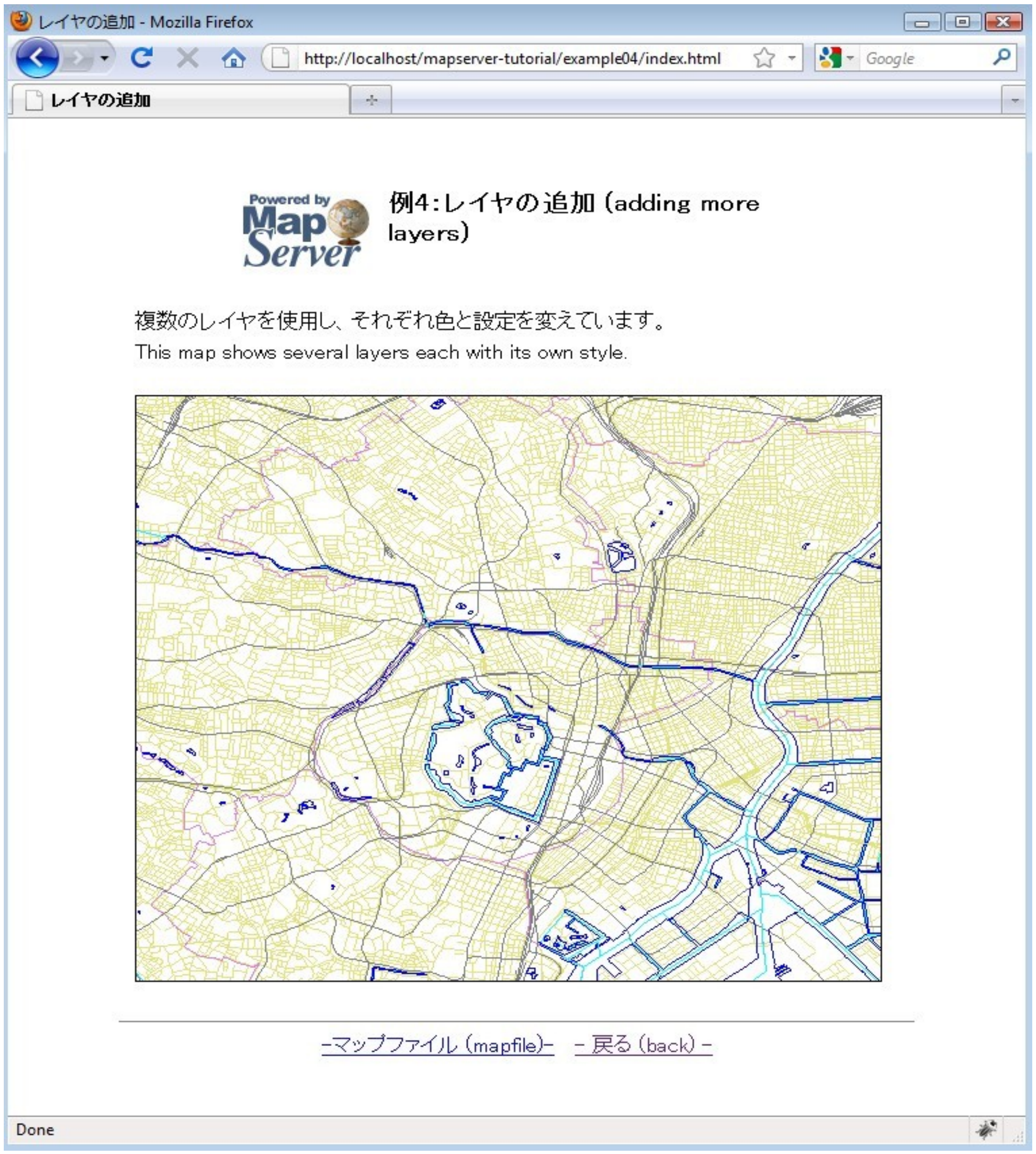


This mapfile contains a FILTER parameter within the LAYER object.

FILTER is used to perform data filtering (before evaluating a CLASS's EXPRESSION). In this case, the filter executed is for when the *KOKUBANGO* field is not equal to 0, and the *YURYO* field is not equal to “—L—¿”.

Exercise 4: Adding More Layers

This example consists of a map containing many layers.


レイヤの追加 - Mozilla Firefox
http://localhost/mapserver-tutorial/example04/index.html
Google
レイヤの追加
Done

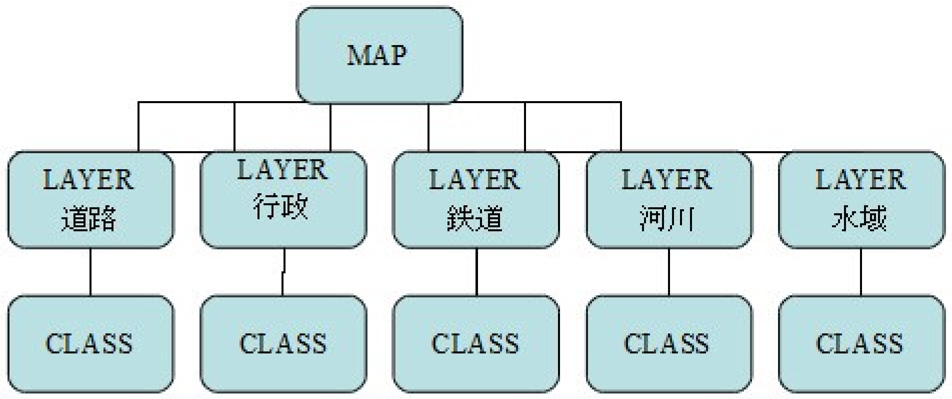
例4:レイヤの追加 (adding more layers)

複数のレイヤを使用し、それぞれ色と設定を変えています。
This map shows several layers each with its own style.

-マップファイル (mapfile)- - 戻る (back) -

The mapfile used in this exercise is:

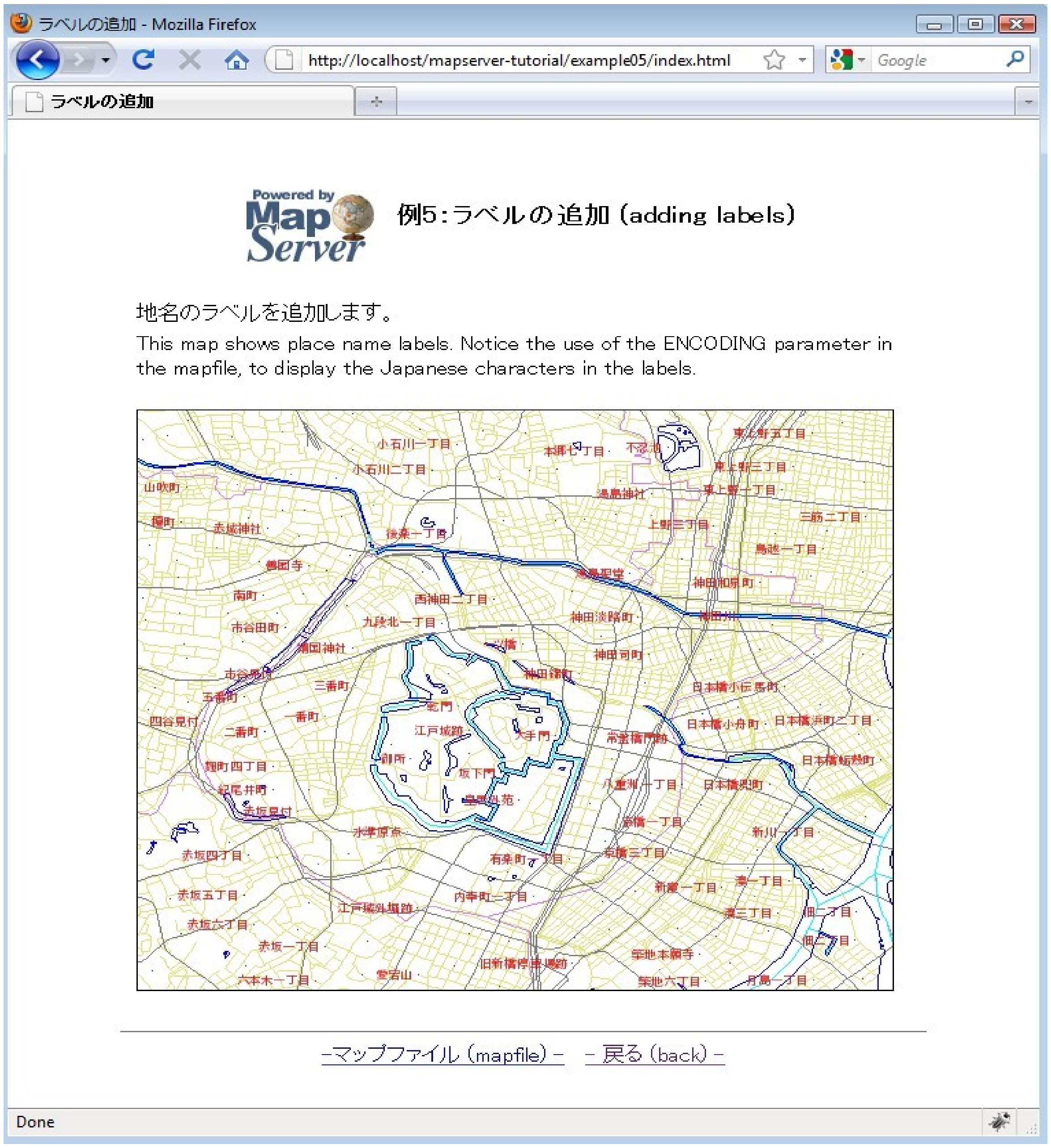
```
MAP
  IMAGETYPE   PNG
  EXTENT      139.70668 35.66589 139.80472 35.72194
  SIZE        550 450
  IMAGECOLOR  255 255 255
  SHAPEPATH   "../data"
  LAYER
    NAME "‶¹˜H"
    DATA dourokukan
    STATUS DEFAULT
    TYPE LINE
    CLASS
      NAME "‶¹˜H"
      STYLE
        COLOR 227 227 127
      END
    END
  END
  LAYER
    NAME "s"
    DATA gyouseikai
    STATUS DEFAULT
    TYPE LINE
    CLASS
      NAME "s"
      STYLE
        COLOR 227 127 227
      END
    END
  END
  LAYER
    NAME "‶S‶¹"
    DATA tetsudokukan
    STATUS DEFAULT
    TYPE LINE
    CLASS
      NAME "“S“¹"
      STYLE
        COLOR 128 128 128
      END
    END
  END
  LAYER
    NAME "‰Íì"
    DATA kasenkukan
    STATUS DEFAULT
    TYPE LINE
    CLASS
      NAME "‰Íì"
      STYLE
        COLOR 0 255 255
      END
    END
  END
  LAYER
    NAME "…ˆæ"
    DATA suiikikai
    STATUS DEFAULT
    TYPE LINE
    CLASS
      NAME "…ˆæ"
      STYLE
        COLOR 0 0 255
      END
    END
  END
END
```



This mapfile example contains 5 LAYERs, each with its own CLASS object. It is important to note that MapServer draws the layers as it reads the .map file, from top to bottom. This means that a layer at the end of the mapfile will appear on top of all of the other layers in the output image.

Exercise 5: Adding Labels

This example shows how to add labels to a layer.



ラベルの追加 - Mozilla Firefox
http://localhost/mapserver-tutorial/example05/index.html
Google
ラベルの追加
Powered by MapServer
Done

例5：ラベルの追加 (adding labels)

地名のラベルを追加します。

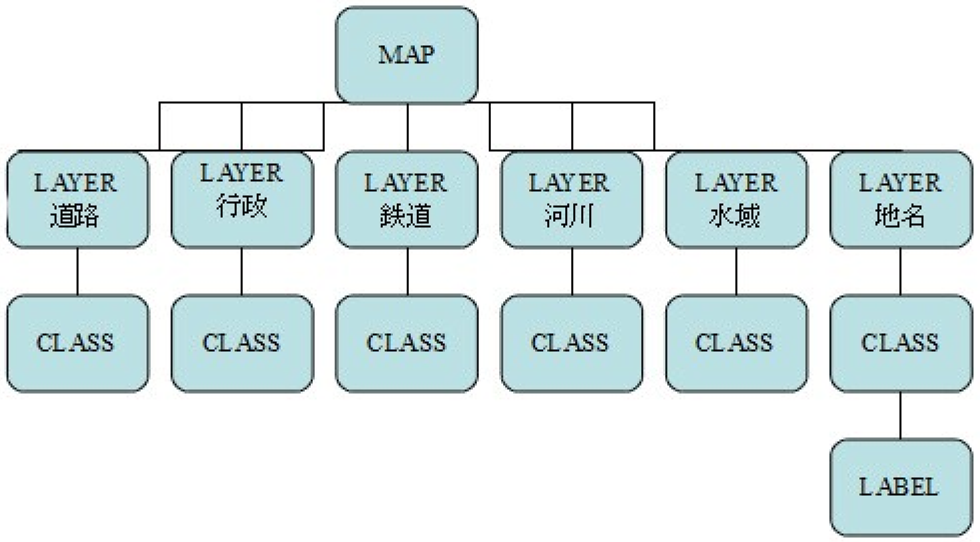
This map shows place name labels. Notice the use of the ENCODING parameter in the mapfile, to display the Japanese characters in the labels.

–マップファイル (mapfile) – – 戻る (back) –

The mapfile used in this exercise is:

```
MAP
  IMAGETYPE   PNG
  EXTENT      139.72520 35.67139 139.78845 35.70731
  SIZE        550 450
  IMAGECOLOR  255 255 255
  SHAPEPATH   "../data"
  FONTSET     ../fonts/fonts.txt
  LAYER
    NAME "“¹~H"
    DATA dourokukan
    STATUS DEFAULT
    TYPE LINE
    CLASS
      NAME "“¹~H"
      STYLE
        COLOR 227 227 127
      END
    END
  END
  LAYER
    NAME "s"
    DATA gyouseikai
    STATUS DEFAULT
    TYPE LINE
    CLASS
      NAME "s"
      STYLE
        COLOR 227 127 227
      END
    END
  END
  LAYER
    NAME "“S“¹"
    DATA tetsudokukan
    STATUS DEFAULT
    TYPE LINE
    CLASS
      NAME "“S“¹"
      STYLE
        COLOR 128 128 128
      END
    END
  END
  LAYER
    NAME "‰Íì"
    DATA kasenkukan
    STATUS DEFAULT
    TYPE LINE
    CLASS
      NAME "‰Íì"
      STYLE
        COLOR 0 255 255
      END
    END
  END
  LAYER
    NAME "…ˆæ"
    DATA suiikikai
    STATUS DEFAULT
    TYPE LINE
    CLASS
      NAME "…ˆæ"
      STYLE
        COLOR 0 0 255
      END
    END
  END
```

```
  LAYER
    NAME "′n−¼"
    DATA chimei
    STATUS DEFAULT
    TYPE POINT
    LABELITEM "NAMAE"
    CLASS
      NAME "′n−¼"
      STYLE
        COLOR 10 100 100
      END
      LABEL
        TYPE TRUETYPE
        FONT pgothic
        COLOR 220 20 20
        SIZE 7
        POSITION CL
        PARTIALS FALSE
        BUFFER 3
        ENCODING "SHIFT_JIS"
      END
    END
  END
END
```



This mapfile contains several layers, with one point layer that uses a LABEL object to show place names. The mapfile used in this example contains several important parameters:

FONTSET
Specify the full path to the font list file. This file defines the available fonts and their aliases. In this case the FONTSET file contains the following (alias, followed by filename):

gothic	ipag.ttf
pgothic	ipagp.ttf
uigothic	ipagui.ttf
mincho	ipam.ttf
pmincho	ipamp.ttf

LABELITEM
Specifies the field to be used for the label values; in this example, the field "NAMAE" contains the label values.

LABEL
Indicates the beginning of the LABEL object.

TYPE
Specify the font type to use. MapServer accepts both BITMAP or TRUETYPE here, but TrueType fonts are scalable through MapServer and are recommended.

FONT
Specify the font to be used. The value specified here is the alias used for the font in the FONTSET.

COLOR
Specify the label text color.

SIZE
If you use TrueType fonts, this is the size of the font specified in pixels. If you use Bitmap fonts, you can specify "tiny", "small", "medium", "large", or "giant".

POSITION
The position of point labels. In this example, "CL" is used to specify center&left of the point.

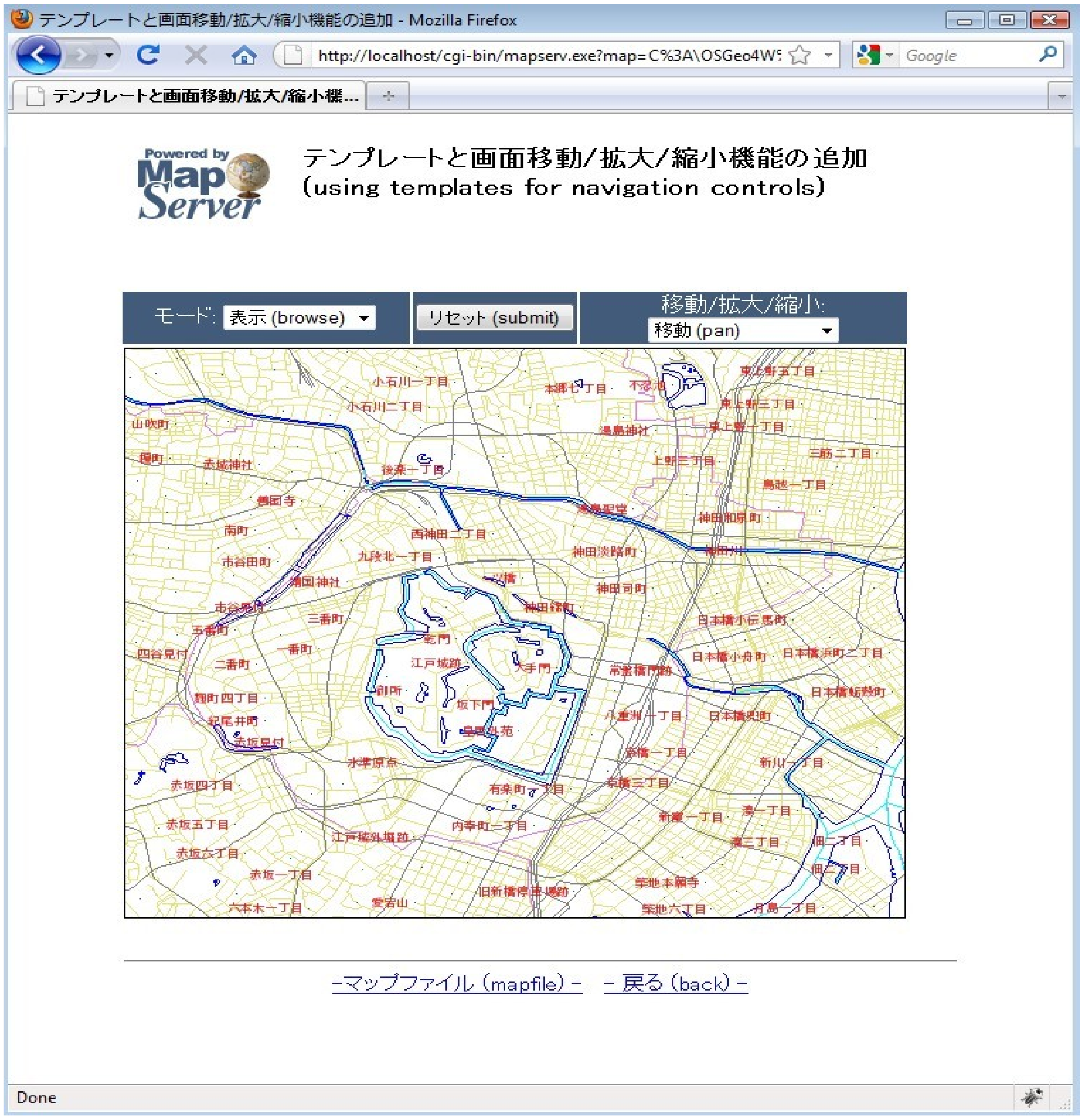
PARTIALS
Specifies whether to include labels that are cut off by the edge of the map. By default, an incomplete label will not be shown. In this example, FALSE means to not show incomplete labels.

BUFFER
The padding of the label in pixels.

ENCODING
The encoding type to use for the label characters. More information about displaying international characters in MapServer can be found at: http://www.mapserver.org/mapfile/labelencoding.html

Exercise 6: Using Templates to Add Navigation Controls

This example shows how use HTML templates to add such tools as navigation to the map interface.


テンプレートと画面移動/拡大/縮小機能の追加 - Mozilla Firefox
http://localhost/cgi-bin/mapserv.exe?map=C%3A\OSGeo4W
テンプレートと画面移動/拡大/縮小機...
テンプレートと画面移動/拡大/縮小機能の追加
(using templates for navigation controls)
モード: 表示 (browse)
リセット (submit)
移動/拡大/縮小:
移動 (pan)
-マップファイル (mapfile) -
- 戻る (back) -
Done

The various tools located above the map image (such as zoom-in or zoom-out) are controlled through an HTML template. The template used in this example follows:

```
<!-- MapServer Template -->
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
<title>ƒeƒ"ƒvƒŒ[ƒg‚Æ‰æ–Ê^Ú"®/Šg'å/k¬‹@"\‚Ì'Ç‰Á</title>
</head>
<body BGCOLOR="white" TEXT="#000000" LINK="#0000FF" VLINK="#0000FF">
<table width="601" align="center" border="0">
<tr>
 <td width="593" align=center>
  <table cellpadding=10 >
  <tr>
    <td align=left><img src="/mapserver-tutorial/img/powered_mapserver.jpg"></td>
    <td align=left><h3>ƒeƒ"ƒvƒŒ[ƒg‚Æ‰æ–Ê^Ú"®/Šg'å/k¬‹@"\‚Ì'Ç‰Á (using templates for
navigation controls)</h3></td>
  </tr>
  </table>
  <br>
<!-- START OF MAPSERVER FORM -->
<form name="mapserv" method="GET" action="/cgi-bin/mapserv.exe">
<!-- HIDDEN MAPSERVER CGI VARIABLES -->
<input type="hidden" name="map" value="[map]">
<input type="hidden" name="imgext" value="[mapext]">
  <tr>
    <td>
       <table width="400" border="0">
      <tr>
        <td bgcolor="#425b7a" width="40%" >
            <!-- SPECIFY MAP MODE -->
           <div align="center"><font color=white>ƒ‚[ƒh: </font>
           <select name="mode">
              <option value="browse">•\Ž¦ (browse)</option>
              <option value="map">'n} (map)</option>
              </select>
           </div>
         </td>
         <td bgcolor="#425b7a" width="15%">
           <!-- FORM SUBMIT BUTTON -->
           <div align="center">
             <input type="submit" name="submit" value="ƒŠƒZƒbƒg (submit)">
           </div>
         </td>
         <td bgcolor="#425b7a" width="45%">
              <!-- ZOOM/PAN CONTROLS -->
            <div align="center"><font color=white>ˆÚ“®/Šg‘å/k¬:</font>
                <select name="zoom">
                  <option value="4" [zoom_4_select]>Šg‘å 4x (zoom-in)</option>
                  <option value="3" [zoom_3_select]>Šg‘å 3x  (zoom-in)</option>
                  <option value="2" [zoom_2_select]>Šg‘å 2x  (zoom-in)</option>
                  <option value="1" [zoom_1_select]>ˆÚ“® (pan)</option>
                  <option value="-2" [zoom_-2_select]>k¬ 2x (zoom-out)</option>
                  <option value="-3" [zoom_-3_select]>k¬ 3x (zoom-out)</option>
                  <option value="-4" [zoom_-4_select]>k¬ 4x (zoom-out)</option>
                </select>
              </div>
         </td>
      </tr>
      <tr>
        <!-- DISPLAY THE MAPSERVER-CREATED MAP IMAGE -->
         <td colspan="3" align="right" valign="top">
            <input type="image" name="img" src="[img]"
                           width="550" height="450" border="1">
         </td>
      </tr>
       </table>
    </td>
  </tr>
</form>
```

```
</table>
<br>
<hr noshade width="80%" size="1" align="center">
<center>
<a href="/mapserver-tutorial/example06/example_map.html">-ƒ}ƒbƒvƒtƒ@ƒCƒ‹ (mapfile)
-</a>&nbsp;&nbsp;
<a href="/mapserver-tutorial/index.html">- –ß,é (back) -</a>
</center>
</body>
</html>
```

The HTML template above contains several MapServer CGI controls; all of the valid MapServer controls are documented at: http://www.mapserver.org/cgi/controls.html

The following are the important points regarding the example template:

<!-- MapServer Template -->

Often referred to as the "MapServer magic string", this string is required at the beginning of every MapServer template file. This is required since MapServer version 5.2.2

Items in square brackets ([map], [mapext], [img], [zoom_n_select]) are used by MapServer; MapServer will replace these with values during run-time.

[Map]
During run-time, this is replaced with the full path to the mapfile (such as "*/OSGeo4W/apps/mapserver-tutorial/example06/example.map*").

[Mapext]
During run-time, this is replaced with the current extents of the map image (such as *"139.725200 35.663485 139.788450 35.715215").*

[Img]
During run-time, this is replaced with the path to the generated map image (such as *"/ms_tmp/MS12821491423356.png").*

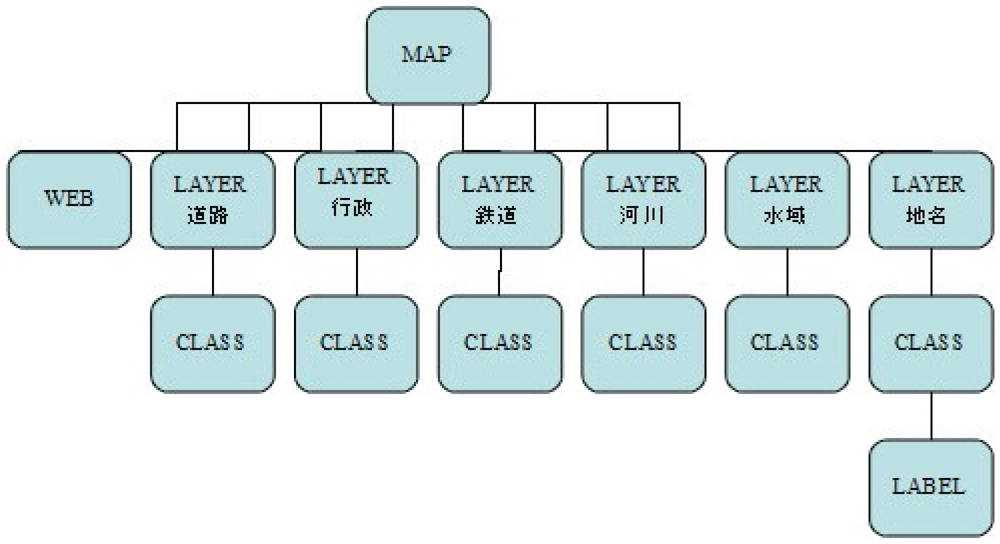
MODE
"Mode" used in the template is "browse", which tells MapServer that a fully interactive map will be generated, or "map" which tells MapServer just to return a static map image to the browser.

[Zoom_n_select]
Represents a zoom of a specified value. Negative values here tell MapServer to zoom out.

The mapfile used in this exercise is:

```
MAP
  IMAGETYPE    PNG
  EXTENT       139.72520 35.67139 139.78845 35.70731
  SIZE         550 450
  IMAGECOLOR   255 255 255
  SHAPEPATH    "../data"
  FONTSET      ../fonts/fonts.txt
  WEB
    TEMPLATE   'example_template.html'
    IMAGEPATH "C:/OSGeo4W/tmp/ms_tmp/"
    IMAGEURL "/ms_tmp/"
  END
  LAYER
    NAME ""¹~H"
    DATA dourokukan
    STATUS DEFAULT
    TYPE LINE
    CLASS
      NAME ""¹~H"
      STYLE
        COLOR 227 227 127
      END
    END
  END
  LAYER
    NAME "s"
    DATA gyouseikai
    STATUS DEFAULT
    TYPE LINE
    CLASS
      NAME "s"
      STYLE
        COLOR 227 127 227
      END
    END
  END
  LAYER
    NAME "“S“¹"
    DATA tetsudokukan
    STATUS DEFAULT
    TYPE LINE
    CLASS
      NAME "“S“¹"
      STYLE
        COLOR 128 128 128
      END
    END
  END
  LAYER
    NAME "‰Íì"
    DATA kasenkukan
    STATUS DEFAULT
    TYPE LINE
    CLASS
      NAME "‰Íì"
      STYLE
        COLOR 0 255 255
      END
    END
  END
  LAYER
    NAME "…ˆæ"
    DATA suiikikai
    STATUS DEFAULT
    TYPE LINE
```

```
    CLASS
      NAME "…^æ"
      STYLE
        COLOR 0 0 255
      END
    END
  END
  LAYER
    NAME "′n−¼"
    DATA chimei
    STATUS DEFAULT
    TYPE POINT
    LABELITEM "NAMAE"
    CLASS
      NAME "′n−¼"
      STYLE
        COLOR 10 100 100
      END
      LABEL
        TYPE TRUETYPE
        FONT pgothic
        COLOR 220 20 20
        SIZE 7
        POSITION CL
        PARTIALS FALSE
        BUFFER 3
        ENCODING "SHIFT_JIS"
      END
    END
  END
END
```



The mapfile in this example includes a WEB object, and its parameters are described below:

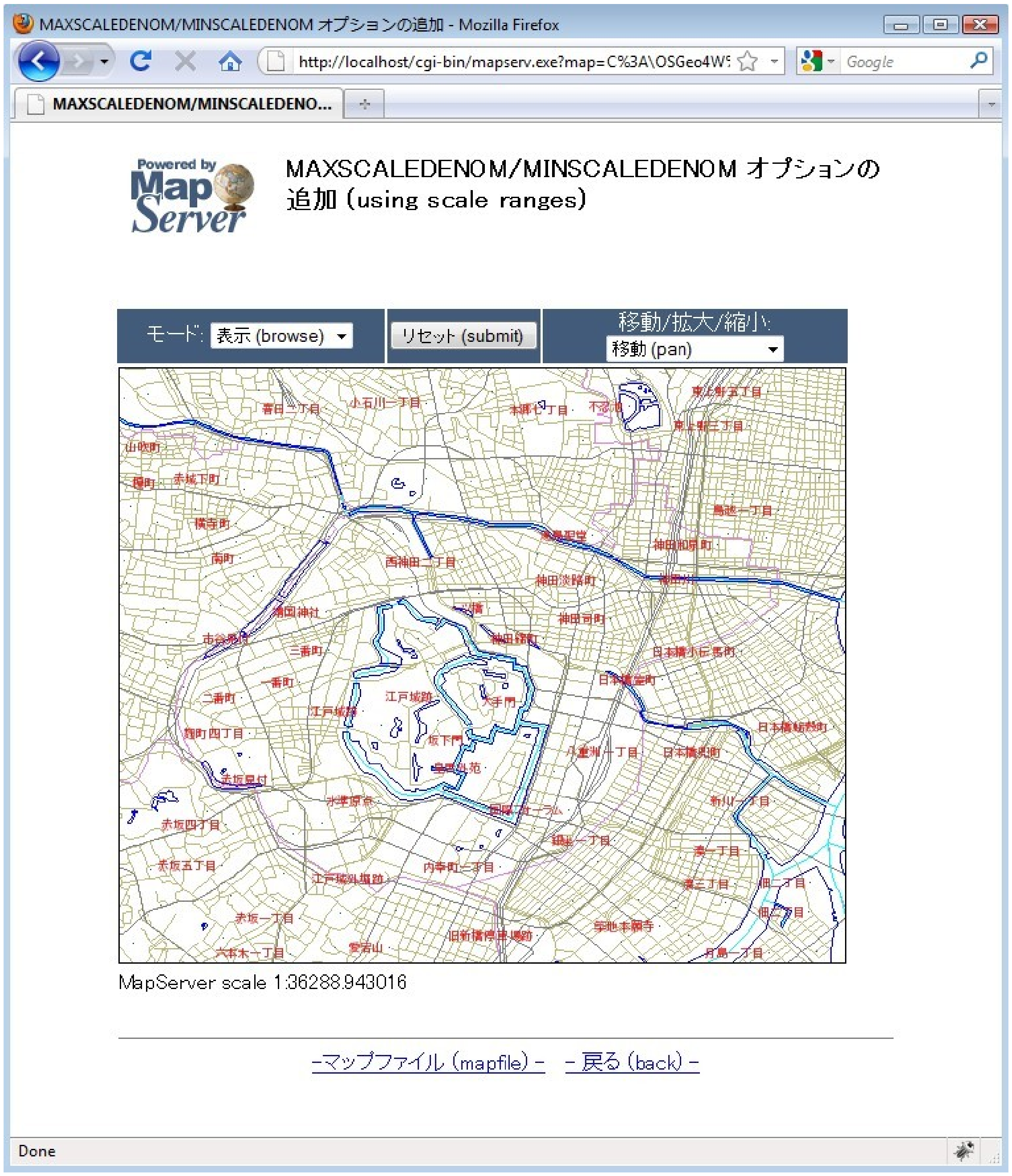
TEMPLATE
The file to use as a template file, relative to the mapfile location. MapServer will replace the controls included in this file with proper values during run-time.

IMAGEPATH
The directory to store the generated map images.

IMAGEURL
The URL to the directory where the generated map images are stored. In this example, OSGeo4W's Apache server is configured for the “ms_tmp” URL.

Exercise 7: Using Scale Ranges

This example shows how use scale ranges at both the MAP level and at the LAYER level in a mapfile.


MAXSCALEDENOM/MINSCALEDENOM オプションの追加 - Mozilla Firefox
http://localhost/cgi-bin/mapserv.exe?map=C%3A\OSGeo4W
MAXSCALEDENOM/MINSCALEDENO...
MAXSCALEDENOM/MINSCALEDENOM オプションの追加 (using scale ranges)
モード: 表示 (browse)
リセット (submit)
移動/拡大/縮小:
移動 (pan)
MapServer scale 1:36288.943016
-マップファイル (mapfile) -
- 戻る (back) -
Done

The mapfile used in this exercise is:

```
MAP
  IMAGETYPE   PNG
  EXTENT      139.72520 35.67139 139.78845 35.70731
  SIZE        550 450
  IMAGECOLOR  255 255 255
  SHAPEPATH   "../data"
  FONTSET     ../fonts/fonts.txt
  UNITS dd
  WEB
    TEMPLATE  'example_template.html'
    IMAGEPATH "C:/OSGeo4W/tmp/ms_tmp/"
    IMAGEURL "/ms_tmp/"
    MINSCALEDENOM  1000
    MAXSCALEDENOM  70000
  END
  LAYER
    NAME "‶¹˜H"
    DATA dourokukan
    STATUS DEFAULT
    TYPE LINE
    MAXSCALEDENOM 50000
    CLASS
      NAME "‶¹˜H"
      STYLE
        COLOR 187 187 127
      END
    END
  END
  LAYER
    NAME "s"
    DATA gyouseikai
    STATUS DEFAULT
    TYPE LINE
    CLASS
      NAME "s"
      STYLE
        COLOR 227 127 227
      END
    END
  END
  LAYER
    NAME "“S“¹"
    DATA tetsudokukan
    STATUS DEFAULT
    TYPE LINE
    CLASS
      NAME "“S“¹"
      STYLE
        COLOR 128 128 128
      END
    END
  END
  LAYER
    NAME "‰Íì"
    DATA kasenkukan
    STATUS DEFAULT
    TYPE LINE

    CLASS
      NAME "‰Íì"
      STYLE
        COLOR 0 255 255
      END
    END
  END
  LAYER
    NAME "…ˆæ"
    DATA suiikikai
    STATUS DEFAULT
    TYPE LINE
    CLASS
      NAME "…ˆæ"
      STYLE
        COLOR 0 0 255
      END
    END
  END
  LAYER
    NAME "’n–¼"
    DATA chimei
    STATUS DEFAULT
    TYPE POINT
    LABELITEM "NAMAE"
    CLASS
      NAME "’n–¼"
      STYLE
        COLOR 10 100 100
      END
      LABEL
        TYPE TRUETYPE
        FONT pgothic
        COLOR 220 20 20
        SIZE 7
        POSITION CL
        PARTIALS FALSE
        BUFFER 5
        ENCODING "SHIFT_JIS"
      END
    END
  END
END
```

As you can see in this mapfile example, the parameters MINSCALEDENOM and MAXSCALEDENOM can be used to specify scale ranges in a mapfile.

MINSCALEDENOM

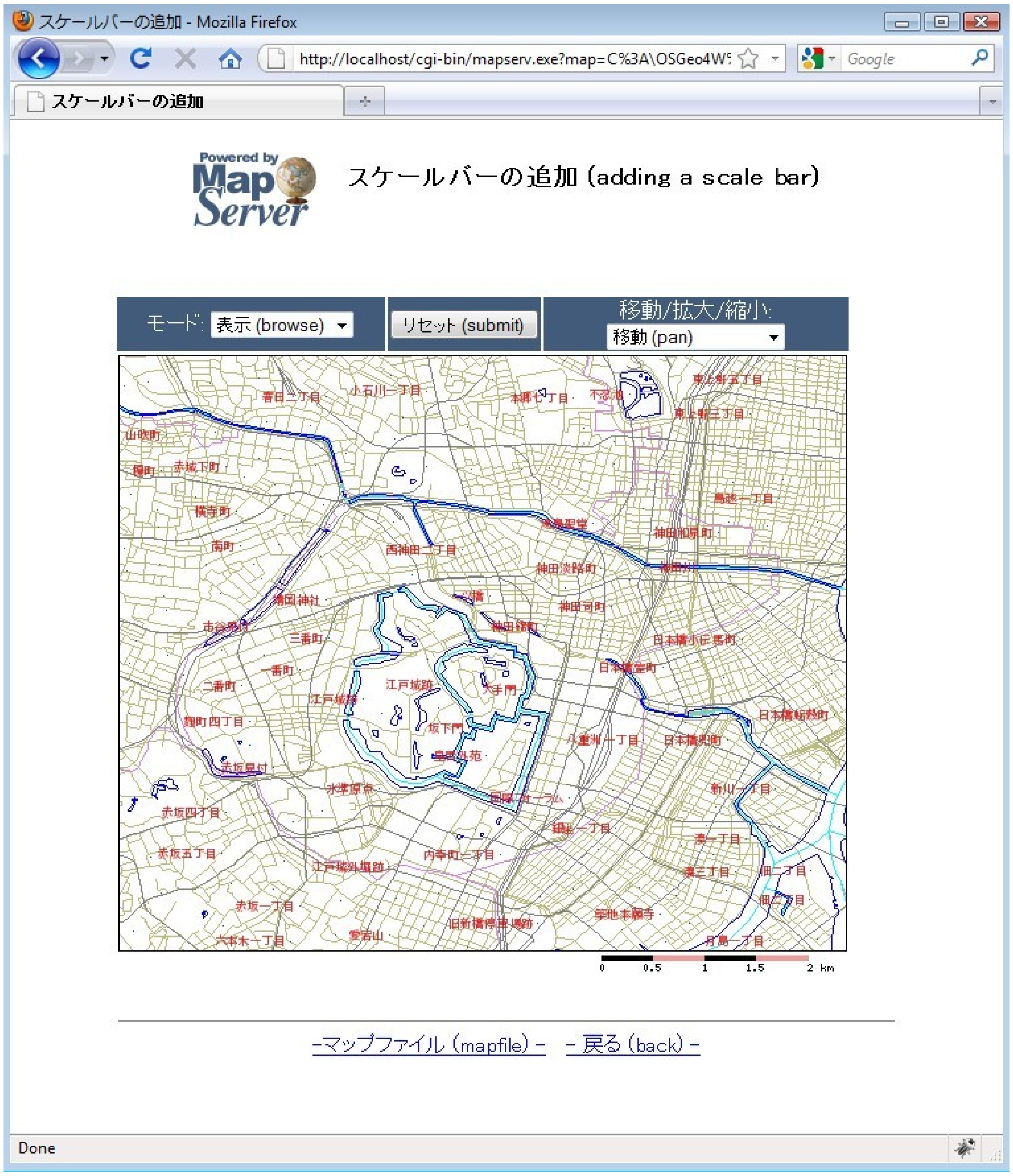
The minimum scale at which to draw this layer. In this example, a MINSCALEDENOM is set in the WEB object, with a value of 1,000 – which means that users will not be able to zoom in past a scale of 1,000.

MAXSCALEDENOM

The maximum scale at which to draw this layer. In this example, a MAXSCALEDENOM is set in the WEB object, with a value of 70,000 – which means that users will not be able to zoom outpast a scale of 70,000. A MAXSCALEDENOM of 50,000 is also set in this example in the *dourokukan* layer – which means that this layer will not be displayed past a scale of 50,000.

Exercise 8: Adding a Scale Bar

This example shows how to add a scale bar through the use of an HTML template.

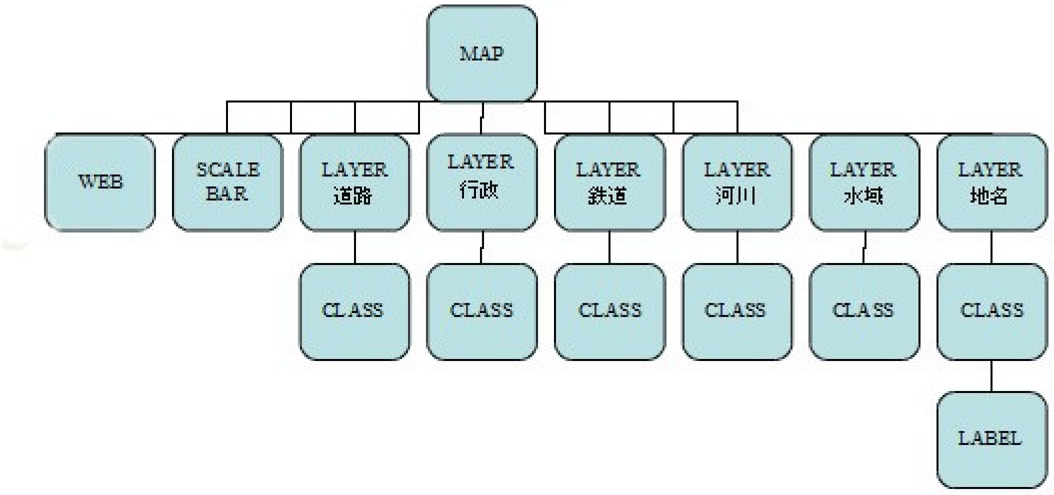

スケールバーの追加 - Mozilla Firefox
http://localhost/cgi-bin/mapserv.exe?map=C%3A\OSGeo4W
スケールバーの追加
スケールバーの追加 (adding a scale bar)
モード: 表示 (browse)
リセット (submit)
移動/拡大/縮小:
移動 (pan)
0 0.5 1 1.5 2 km
-マップファイル (mapfile) -
- 戻る (back) -
Done

The mapfile used in this exercise is:

```
MAP
  IMAGETYPE    PNG
  EXTENT       139.72520 35.67139 139.78845 35.70731
  SIZE         550 450
  IMAGECOLOR   255 255 255
  SHAPEPATH    "../data"
  FONTSET      ../fonts/fonts.txt
  UNITS dd
  WEB
    TEMPLATE  'example_template.html'
    IMAGEPATH "C:/OSGeo4W/tmp/ms_tmp/"
    IMAGEURL "/ms_tmp/"
    MINSCALEDENOM  1000
    MAXSCALEDENOM  70000
  END
  SCALEBAR
    UNITS kilometers
    BACKGROUNDCOLOR 250 150 150
    COLOR 0 0 0
    TRANSPARENT on
    STYLE 0
    STATUS on
    LABEL
      COLOR 0 0 0
      SIZE tiny
    END
  END
  LAYER
    NAME "‶¹˜H"
    DATA dourokukan
    STATUS DEFAULT
    TYPE LINE
    MAXSCALEDENOM 50000
    CLASS
      NAME "“¹˜H"
      STYLE
        COLOR 187 187 127
      END
    END
  END
  LAYER
    NAME "s"
    DATA gyouseikai
    STATUS DEFAULT
    TYPE LINE

    CLASS
      NAME "s"
      STYLE
        COLOR 227 127 227
      END
    END
  END
  LAYER
    NAME "“S“¹"
    DATA tetsudokukan
    STATUS DEFAULT
    TYPE LINE
    CLASS
      NAME "“S“¹"
      STYLE
        COLOR 128 128 128
      END
    END
  END
  LAYER
    NAME "‰Íì"
    DATA kasenkukan
    STATUS DEFAULT
    TYPE LINE
    CLASS
      NAME "‰Íì"
      STYLE
        COLOR 0 255 255
      END
    END
  END
  LAYER
    NAME "…ˆæ"
    DATA suiikikai
    STATUS DEFAULT
    TYPE LINE
    CLASS
      NAME "…ˆæ"
      STYLE
        COLOR 0 0 255
      END
    END
  END
  LAYER
    NAME "’n–¼"
    DATA chimei
    STATUS DEFAULT
    TYPE POINT
    LABELITEM "NAMAE"
    CLASS
      NAME "’n–¼"
      STYLE
        COLOR 10 100 100
      END
      LABEL
```

```
        TYPE TRUETYPE
        FONT pgothic
        COLOR 220 20 20
        SIZE 7
        POSITION CL
        PARTIALS FALSE
        BUFFER 5
        ENCODING "SHIFT_JIS"
      END
    END
  END
END
```



In this example, a SCALEBAR object is added to the mapfile, and its parameters are described next:

SCALEBAR
Marks the start of the SCALEBAR object.

UNITS
Specify the units of scale bar.

BACKGROUNDCOLOR
Color to use for the scalebar background.

COLOR
Color to use for the scalebar bars.

TRANSPARENT
Specifies whether the background color for the scalebar should be transparent.

STYLE
Specifies the style of scale bar.

STATUS
Specifies whether to draw the scale bar.

The template file used in this exercise follows:

```
<!-- MapServer Template -->
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
<title>ƒXƒP[ƒ‹ƒo[‚Ì’Ç‰Á</title>
</head>
<body BGCOLOR="white" TEXT="#000000" LINK="#0000FF" VLINK="#0000FF">
<table width="601" align="center" border="0">
<tr>
 <td width="593" align=center>
  <table cellpadding=10 >
  <tr>
    <td align=left><img src="/mapserver-tutorial/img/powered_mapserver.jpg"></td>
    <td align=left><h3>ƒXƒP[ƒ‹ƒo[‚Ì’Ç‰Á (adding a scale bar)</h3></td>
  </tr>
  </table>
  <br>
<!-- START OF MAPSERVER FORM -->
<form name="mapserv" method="GET" action="/cgi-bin/mapserv.exe">
<!-- HIDDEN MAPSERVER CGI VARIABLES -->
<input type="hidden" name="map" value="[map]">
<input type="hidden" name="imgext" value="[mapext]">
  <tr>
```

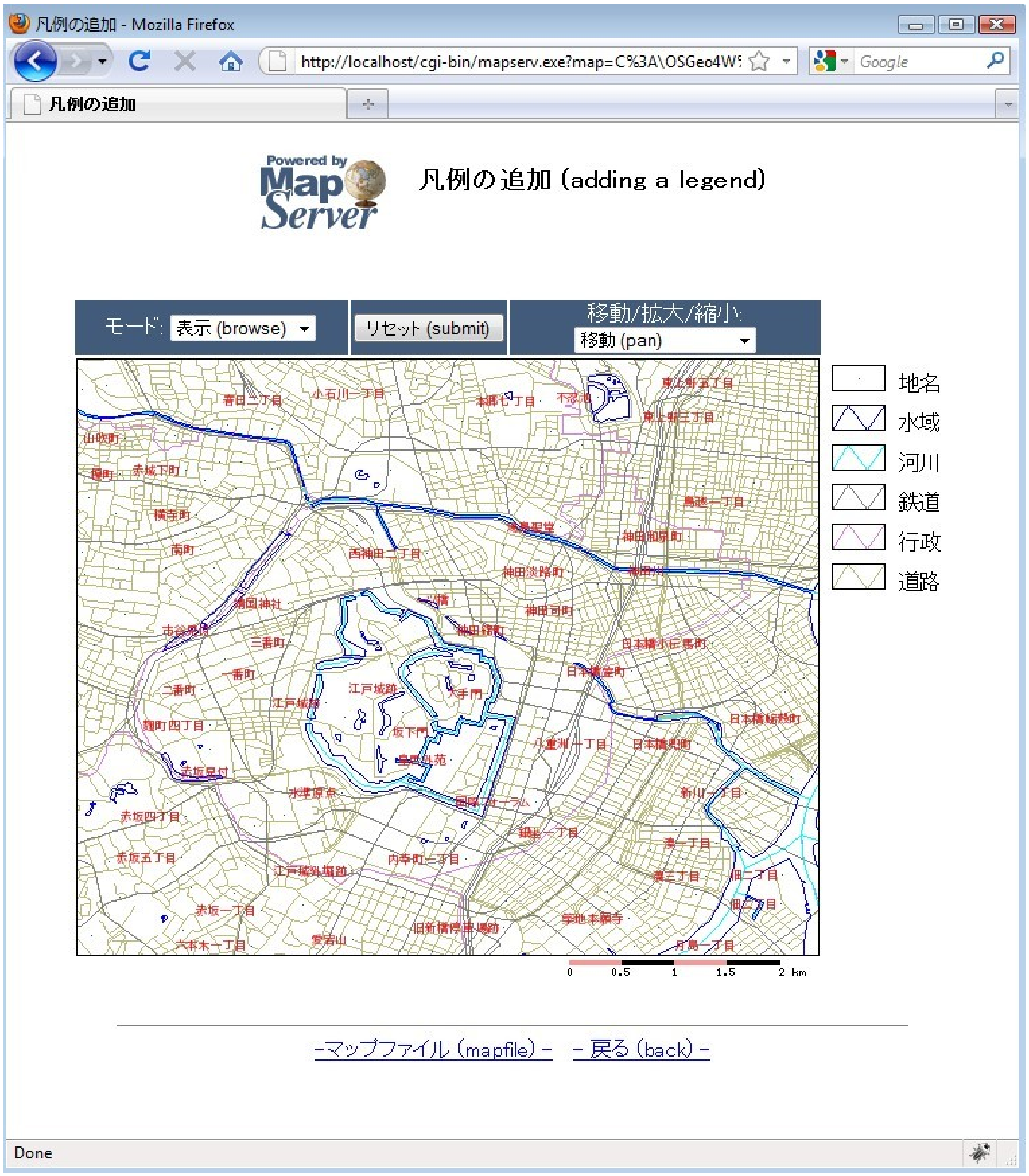
```
    <td>
       <table width="400" border="0">
      <tr>
        <td bgcolor="#425b7a" width="40%" >
            <!-- SPECIFY MAP MODE -->
           <div align="center"><font color=white>ƒ‚[ƒh: </font>
           <select name="mode">
              <option value="browse">•\Ž¦ (browse)</option>
              <option value="map">’n} (map)</option>
              </select>
           </div>
         </td>
         <td bgcolor="#425b7a" width="15%">
           <!-- FORM SUBMIT BUTTON -->
           <div align="center">
             <input type="submit" name="submit" value="ƒŠƒZƒbƒg (submit)">
           </div>
         </td>
         <td bgcolor="#425b7a" width="45%">
              <!-- ZOOM/PAN CONTROLS -->
            <div align="center"><font color=white>ˆÚ“®/Šg‘å/k¬:</font>
                <select name="zoom">
                  <option value="4" [zoom_4_select]>Šg‘å 4x (zoom-in)</option>
                  <option value="3" [zoom_3_select]>Šg‘å 3x (zoom-in)</option>
                  <option value="2" [zoom_2_select]>Šg‘å 2x (zoom-in)</option>
                  <option value="1" [zoom_1_select]>ˆÚ“® (pan)</option>
                  <option value="-2" [zoom_-2_select]>k¬ 2x (zoom-out)</option>
                  <option value="-3" [zoom_-3_select]>k¬ 3x (zoom-out)</option>
                  <option value="-4" [zoom_-4_select]>k¬ 4x (zoom-out)</option>
                </select>
              </div>
         </td>
      </tr>
      <tr>
```

```
        <!-- DISPLAY THE MAPSERVER-CREATED MAP IMAGE -->
         <td colspan="3" align="right" valign="top">
            <input type="image" name="img" src="[img]"
                           width="550" height="450" border="1">
        <!-- DISPLAY THE SCALEBAR IMAGE -->
              <img src="[scalebar]" border="0">
         </td>
      </tr>
       </table>
    </td>
  </tr>
</form>
</table>
<br>
<hr noshade width="80%" size="1" align="center">
<center>
<a href="/mapserver-tutorial/example08/example_map.html">-ƒ}ƒbƒvƒtƒ@ƒCƒ‹  (mapfile)
-</a>&nbsp;&nbsp;
<a href="/mapserver-tutorial/index.html">- ß,é (back) -</a>
</center>
</body>
</html>
```

The HTML template above contains the [scalebar] control, that is replaced by MapServer during run-time with a scalebar image.

Exercise 9: Adding a Legend

This example shows how to add a legend through the use of an HTML template.



In order to display a legend, the following is required:

1) the mapfile must contain a LEGEND object
2) each CLASS object in the mapfile must have a NAME parameter
3) the HTML template must contain a [legend] control.

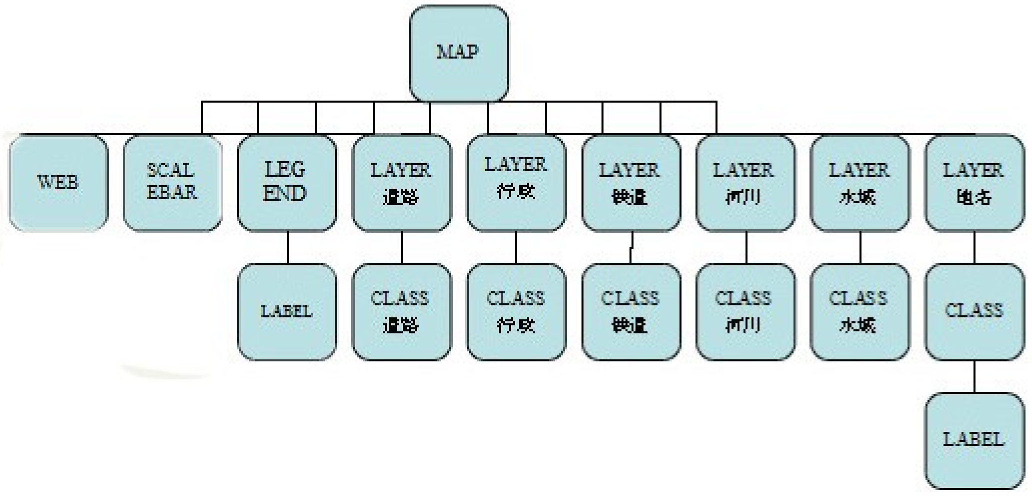
The mapfile used in this exercise is:

```
MAP
  IMAGETYPE   PNG
  EXTENT      139.72520 35.67139 139.78845 35.70731
  SIZE        550 450
  IMAGECOLOR  255 255 255
  SHAPEPATH   "../data"
  FONTSET     ../fonts/fonts.txt
  UNITS dd
  WEB
    TEMPLATE  'example_template.html'
    IMAGEPATH "C:/OSGeo4W/tmp/ms_tmp/"
    IMAGEURL "/ms_tmp/"
    MINSCALEDENOM  1000
    MAXSCALEDENOM  70000
  END
  SCALEBAR
    UNITS kilometers
    COLOR 250 150 150
    BACKGROUNDCOLOR 0 0 0
    TRANSPARENT on
    STYLE 0
    STATUS on
    LABEL
      COLOR 0 0 0
      SIZE tiny
    END
  END
```

```
LEGEND
  KEYSIZE 40 20
  KEYSPACING 10 10
  OUTLINECOLOR 0 0 0
  IMAGECOLOR 255 255 255
  LABEL
    TYPE TRUETYPE
    FONT pmincho
    COLOR 0 0 0
    SIZE 12
    POSITION CL
    PARTIALS FALSE
    BUFFER 3
    ENCODING "SHIFT_JIS"
  END
  STATUS ON
END
LAYER
  NAME "‶¹~H"
  DATA dourokukan
  STATUS DEFAULT
  TYPE LINE
  MAXSCALEDENOM 50000
  CLASS
    NAME "‶¹~H"
    STYLE
      COLOR 187 187 127
    END
  END
END
LAYER
  NAME "s"
  DATA gyouseikai
  STATUS DEFAULT
  TYPE LINE
  CLASS
    NAME  "s"
    STYLE
      COLOR 227 127 227
    END
  END
END
LAYER
  NAME "“S“¹"
  DATA tetsudokukan
  STATUS DEFAULT
  TYPE LINE
  CLASS
    NAME "“S“¹"
    STYLE
      COLOR 128 128 128
    END
  END
END
LAYER
  NAME "‰Íì"
  DATA kasenkukan
  STATUS DEFAULT
  TYPE LINE

  CLASS
    NAME "‰Íì"
    STYLE
      COLOR 0 255 255
    END
  END
END
LAYER
```

```
    NAME "…^æ"
    DATA suiikikai
    STATUS DEFAULT
    TYPE LINE
    CLASS
      NAME "…^æ"
      STYLE
        COLOR 0 0 255
      END
    END
  END
  LAYER
    NAME "’n−¼"
    DATA chimei
   STATUS DEFAULT
    TYPE POINT
    LABELITEM "NAMAE"
    CLASS
      NAME "’n−¼"
      STYLE
        COLOR 10 100 100
      END
      LABEL
        TYPE TRUETYPE
        FONT pgothic
        COLOR 220 20 20
        SIZE 7
        POSITION CL
        PARTIALS FALSE
        BUFFER 5
        ENCODING "SHIFT_JIS"
      END
    END
  END
END
```



As you can see in the mapfile, a SCALEBAR object is added, and its parameters are described next:

LEGEND
Marks the start of the LEGEND object.

KEYSIZE
Specifies the size in pixels of the key image.

KEYSPACING
Specify the space between the key image and the label, in pixels.

OUTLINECOLOR
Specify the line color of the key image.

IMAGECOLOR
Specifies the background color of the legend.

STATUS
Specifies whether to display the legend.

The template file used in this exercise follows:

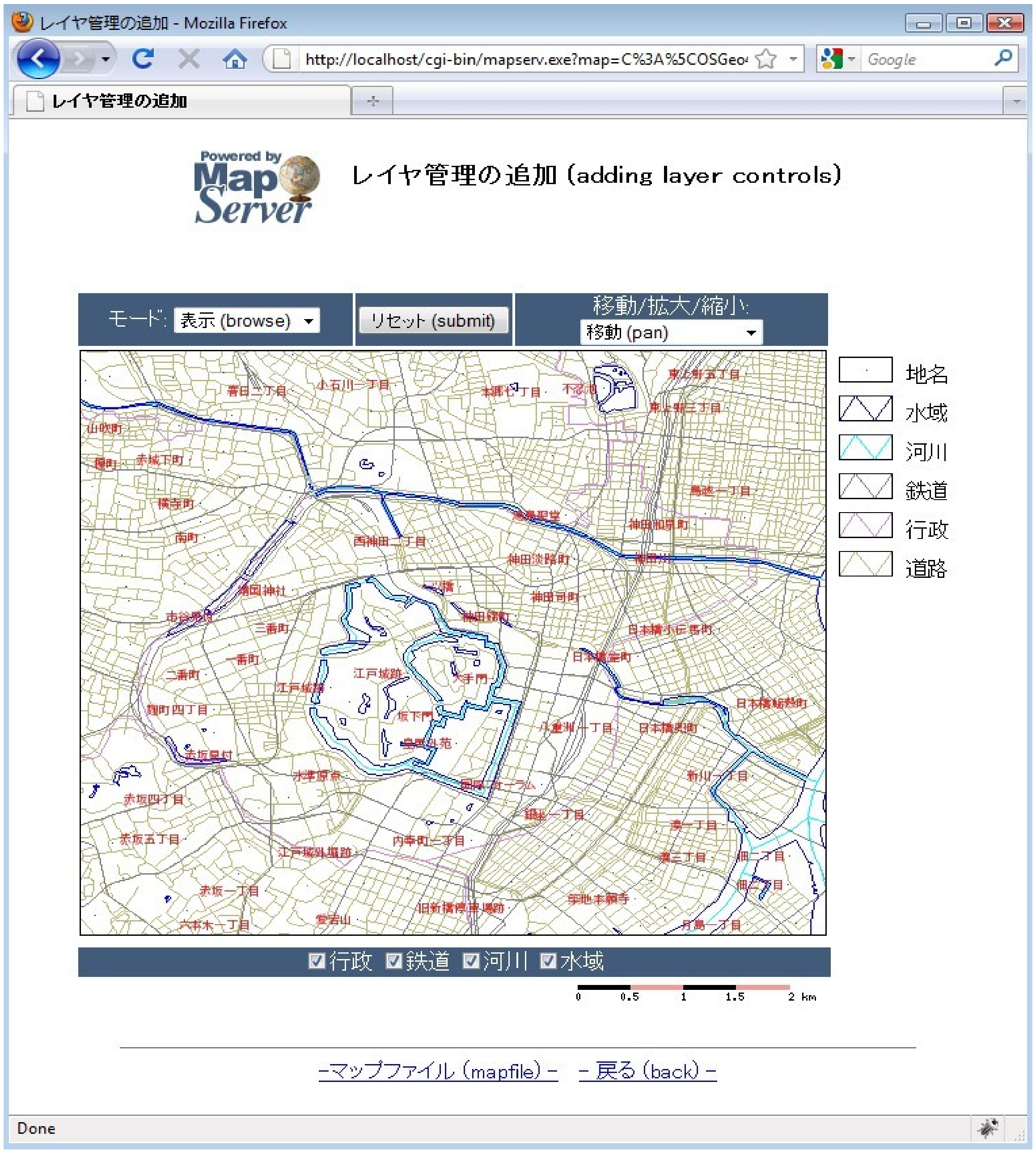
```
<!-- MapServer Template -->
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
<title>–}—á‚Ì’Ç‰Á</title>
</head>
<body BGCOLOR="white" TEXT="#000000" LINK="#0000FF" VLINK="#0000FF">
<table width="601" align="center" border="0">
<tr>
 <td width="593" align=center>
  <table cellpadding=10 >
  <tr>
    <td align=left><img src="/mapserver-tutorial/img/powered_mapserver.jpg"></td>
    <td align=left><h3>–}—á‚Ì’Ç‰Á (adding a legend)</h3></td>
  </tr>
  </table>
  <br>
<!-- START OF MAPSERVER FORM -->
<form name="mapserv" method="GET" action="/cgi-bin/mapserv.exe">
<!-- HIDDEN MAPSERVER CGI VARIABLES -->
<input type="hidden" name="map" value="[map]">
<input type="hidden" name="imgext" value="[mapext]">
  <tr>
    <td>
       <table width="400" border="0">
      <tr>
        <td bgcolor="#425b7a" width="40%" >
            <!-- SPECIFY MAP MODE -->
           <div align="center"><font color=white>ƒ‚[ƒh: </font>
           <select name="mode">
              <option value="browse">•\Ž¦ (browse)</option>
              <option value="map">’n} (map)</option>
              </select>
           </div>
         </td>
         <td bgcolor="#425b7a" width="15%">
           <!-- FORM SUBMIT BUTTON -->
           <div align="center">
             <input type="submit" name="submit" value="ƒŠƒZƒbƒg (submit)">
           </div>
         </td>
         <td bgcolor="#425b7a" width="45%">
              <!-- ZOOM/PAN CONTROLS -->
            <div align="center"><font color=white>ˆÚ“®/Šg'å/k¬:</font>
                <select name="zoom">
                  <option value="4" [zoom_4_select]>Šg'å 4x (zoom-in)</option>
                  <option value="3" [zoom_3_select]>Šg'å 3x (zoom-in)</option>
                  <option value="2" [zoom_2_select]>Šg'å 2x (zoom-in)</option>
                  <option value="1" [zoom_1_select]>ˆÚ“® (pan)</option>
                  <option value="-2" [zoom_-2_select]>k¬ 2x (zoom-out)</option>
                  <option value="-3" [zoom_-3_select]>k¬ 3x (zoom-out)</option>
                  <option value="-4" [zoom_-4_select]>k¬ 4x (zoom-out)</option>
                </select>
              </div>
         </td>
      </tr>
      <tr>
        <!-- DISPLAY THE MAPSERVER-CREATED MAP IMAGE -->
         <td colspan="3" align="right" valign="top">
            <input type="image" name="img" src="[img]"
                           width="550" height="450" border="1">
        <!-- DISPLAY THE SCALEBAR IMAGE -->
              <img src="[scalebar]" border="0">
         </td>
           <td valign="top">
        <!-- DISPLAY THE LEGEND IMAGE -->
              <img src="[legend]">
           </td>
```

```
      </tr>
       </table>
    </td>
  </tr>
</form>
</table>
<br>
<hr noshade width="80%" size="1" align="center">
<center>
<a href="/mapserver-tutorial/example09/example_map.html">-ƒ}ƒbƒvƒtƒ@ƒCƒ‹  (mapfile
-</a>&nbsp;&nbsp;
<a href="/mapserver-tutorial/index.html">- –ß‚é (back) -</a>
</center>
</body>
</html>
```

The above template includes the [legend] control which MapServer will replace at run-time with a legend image.

Exercise 10: Adding Layer Controls

This example shows how to add the ability to turn layers on and off through the use of an HTML template.



Below the map image you will notice some layer control checkboxes.

To allow the user to turn on/off layers, in the HTML template file you need to add a check box with the name "layer", and the value of each checkbox should be the name of the layer from the mapfile; MapServer will change the layer's STATUS parameter accordingly.

The mapfile used in this exercise is:

```
MAP
  IMAGETYPE   PNG
  EXTENT      139.72520 35.67139 139.78845 35.70731
  SIZE        550 450
  IMAGECOLOR  255 255 255
  SHAPEPATH   "../data"
  FONTSET     ../fonts/fonts.txt
  UNITS dd
  WEB
    TEMPLATE  'example_template.html'
    IMAGEPATH "@osgeo4w@/tmp/ms_tmp/"
    IMAGEURL "/ms_tmp/"
    MINSCALEDENOM  1000
    MAXSCALEDENOM  70000
  END
  SCALEBAR
    UNITS kilometers
    BACKGROUNDCOLOR 250 150 150
    COLOR 0 0 0
    TRANSPARENT on
    STYLE 0
    STATUS on
    LABEL
      COLOR 0 0 0
      SIZE tiny
    END
  END
  LEGEND
```

```
    KEYSIZE 40 20

    KEYSPACING 10 10
    OUTLINECOLOR 0 0 0
    IMAGECOLOR 255 255 255
    LABEL
      TYPE TRUETYPE
      FONT pmincho
      COLOR 0 0 0
      SIZE 12
      POSITION CL
      PARTIALS FALSE
      BUFFER 3
      ENCODING "SHIFT_JIS"
    END
    STATUS ON
  END
  LAYER
    NAME "“¹˜H"
    DATA dourokukan
    STATUS DEFAULT
    TYPE LINE
    MAXSCALEDENOM 50000
    CLASS
      NAME "“¹˜H"
      COLOR 187 187 127
    END
  END
  LAYER
    NAME "s"
    DATA gyouseikai
    STATUS ON
    TYPE LINE
    CLASS
      NAME "s"
      STYLE
      COLOR 227 127 227
    END
  END
END
LAYER
  NAME "“S“¹"
  DATA tetsudokukan
  STATUS ON
  TYPE LINE
  CLASS
    NAME "“S“¹"
    STYLE
      COLOR 128 128 128
    END
  END
END
LAYER
  NAME "‰Íì"
  DATA kasenkukan
  STATUS ON
  TYPE LINE
  CLASS
    NAME "‰Íì"
    STYLE
      COLOR 0 255 255
    END
  END
END
LAYER
  NAME "…ˆæ"
  DATA suiikikai
  STATUS ON
  TYPE LINE
  CLASS
```

```
      NAME "…ˆæ"
      STYLE
        COLOR 0 0 255
      END
    END
  END
  LAYER
    NAME "’n–¼"
    DATA chimei
    STATUS DEFAULT
    TYPE POINT
    LABELITEM "NAMAE"
    CLASS
      NAME "’n–¼"
      STYLE
        COLOR 10 100 100
      END
      LABEL
        TYPE TRUETYPE
        FONT pgothic
        COLOR 220 20 20
        SIZE 7
        POSITION CL
        PARTIALS FALSE
        BUFFER 5
        ENCODING "SHIFT_JIS"
      END
    END
  END
END
```

Note the STATUS value for each LAYER. Layers with a status of DEFAULT will always be displayed, but the other layers with status ON will be controlled by the HTML template.

The template file used in this exercise follows:

```
<!-- MapServer Template -->
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
<title>ƒŒƒCƒ„ŠÇ—‚Ì’Ç‰Á</title>
<script type="text/javascript">
  function doSubmit()
  {
    document.mainform.submit();
  }
</script>
</head>
<body BGCOLOR="white" TEXT="#000000" LINK="#0000FF" VLINK="#0000FF">
<table width="601" align="center" border="0">
<tr>
 <td width="593" align=center>
  <table cellpadding=10 >
  <tr>
    <td align=left><img src="/mapserver-tutorial/img/powered_mapserver.jpg"></td>
    <td align=left><h3>ƒŒƒCƒ„ŠÇ—‚Ì’Ç‰Á (adding layer controls)</h3></td>
  </tr>
  </table>
  <br>
<!-- START OF MAPSERVER FORM -->
<form name="mainform" method="GET" action="/cgi-bin/mapserv.exe">
<!-- HIDDEN MAPSERVER CGI VARIABLES -->
<input type="hidden" name="map" value="[map]">
<input type="hidden" name="imgext" value="[mapext]">
<input type="hidden" name="mapext" value="[mapext]">
  <tr>
    <td>
       <table width="400" border="0">
      <tr>
```

```
        <td bgcolor="#425b7a" width="40%" >
            <!-- SPECIFY MAP MODE -->
          <div align="center"><font color=white>ƒ‚[ƒh: </font>
          <select name="mode">
             <option value="browse">•\Ž¦ (browse)</option>
             <option value="map">’n} (map)</option>
             </select>
          </div>
         </td>
         <td bgcolor="#425b7a" width="15%">
          <!-- FORM SUBMIT BUTTON -->
          <div align="center">
            <input type="button" name="submitter" value="ƒŠƒZƒbƒg (submit)"
onclick="doSubmit()">
          </div>
         </td>
         <td bgcolor="#425b7a" width="45%">
              <!-- ZOOM/PAN CONTROLS -->
           <div align="center"><font color=white>ˆÚ“®/Šg‘å/k¬:</font>
               <select name="zoom">
                 <option value="4" [zoom_4_select]>Šg‘å 4x (zoom-in)</option>
                 <option value="3" [zoom_3_select]>Šg‘å 3x (zoom-in)</option>
                 <option value="2" [zoom_2_select]>Šg‘å 2x (zoom-in)</option>
                 <option value="1" [zoom_1_select]>ˆÚ“® (pan)</option>
                 <option value="-2" [zoom_-2_select]>k¬ 2x (zoom-out)</option>
                 <option value="-3" [zoom_-3_select]>k¬ 3x (zoom-out)</option>
                 <option value="-4" [zoom_-4_select]>k¬ 4x (zoom-out)</option>
               </select>
              </div>
         </td>
      </tr>
      <tr>
        <!-- DISPLAY THE MAPSERVER-CREATED MAP IMAGE -->
         <td colspan="3" align="right" valign="top">
           <input type="image" name="img" src="[img]"
```

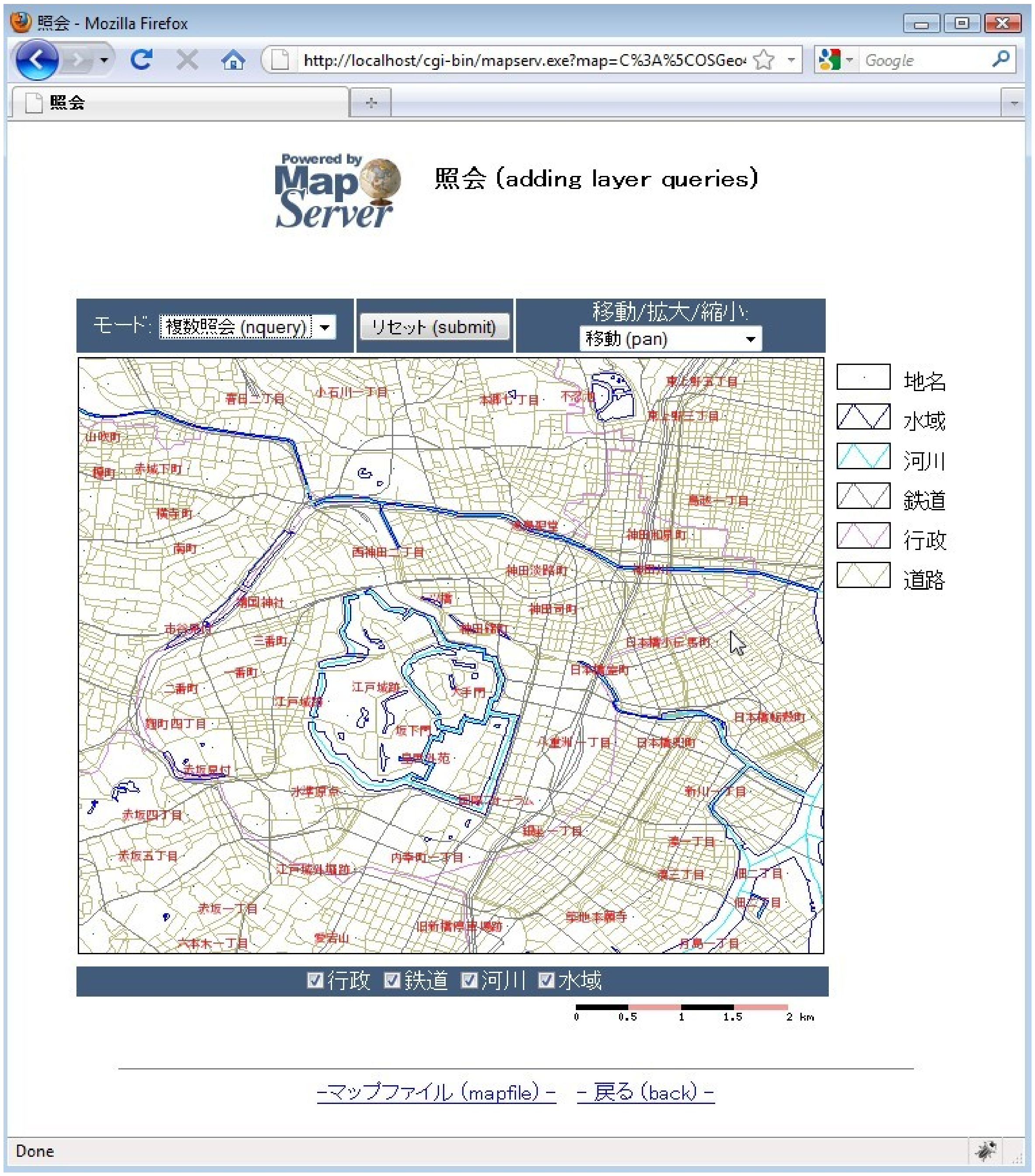
```
                           width="550" height="450" border="1">
         </td>
           <td valign=top >
            <!-- DISPLAY THE LEGEND IMAGE -->
            <img src="[legend]">
           </td>
      </tr>
       </table>
    </td>
  </tr>
  <tr>
    <td>
       <table width=560 border="0">
         <tr>
           <td bgcolor="#425b7a">
             <font color="white">
             <div align="center">
             <input type=checkbox name=layer value="s" [s_check]
onclick="doSubmit()">s
             <input type=checkbox name=layer value=""S"¹" ["S"¹_check]
onclick="doSubmit()">"S"¹
             <input type=checkbox name=layer value="‰Íì" [‰Íì_check]
onclick="doSubmit()">‰Íì
             <input type=checkbox name=layer value="…ˆæ" […ˆæ_check]
onclick="doSubmit()">…ˆæ
             </font>
           </td>
         </tr>
         <tr>
           <td align=right>
              <!-- DISPLAY THE SCALEBAR IMAGE -->
              <img src="[scalebar]" border="0">
           </td>
         </tr>
       </table>
    </td>


  </tr>
</form>
</table>
<br>
<hr noshade width="80%" size="1" align="center">
<center>
<a href="/mapserver-tutorial/example10/example_map.html">-ƒ}ƒbƒvƒtƒ@ƒCƒ‹  (mapfile)
-</a>&nbsp;&nbsp;
<a href="/mapserver-tutorial/index.html">- –ß‚é (back) -</a>
</center>
</body>
</html>
```

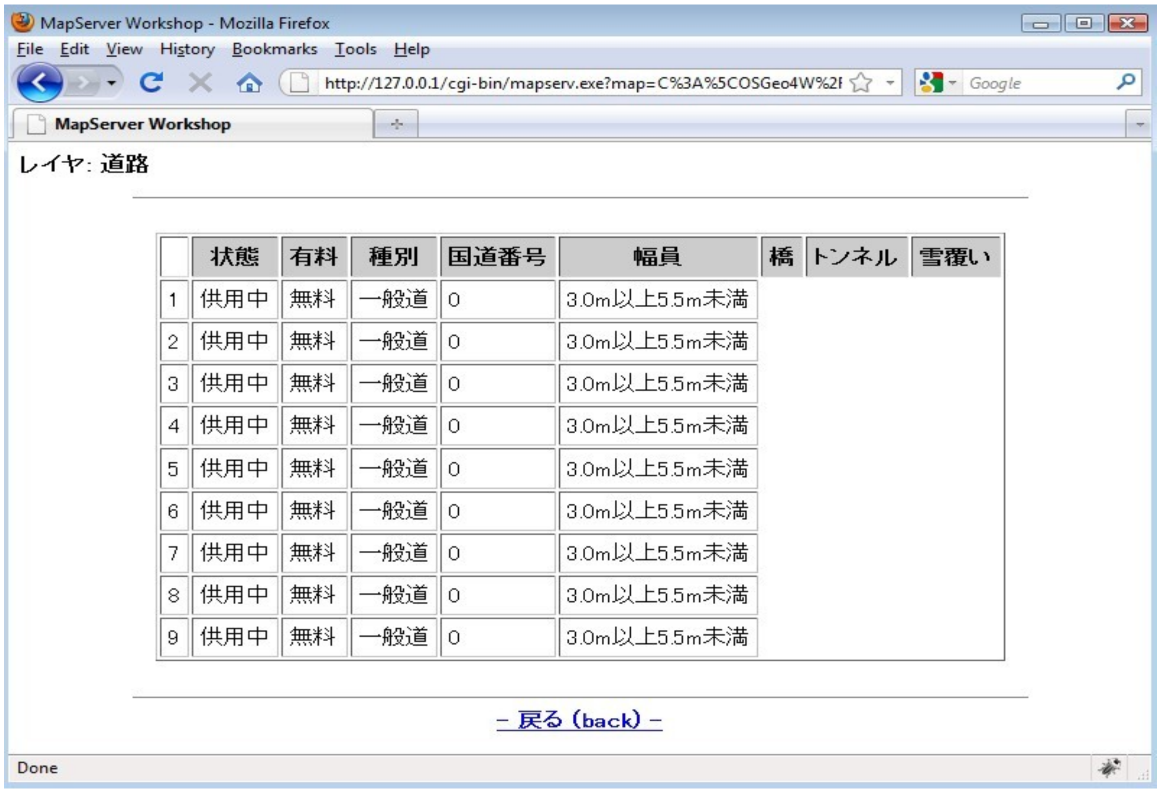
Note the checkbox used to control the layer display. The value of each checkbox is the NAME of the LAYER, as it appears in the mapfile.

Exercise 11: Adding Layer Queries

This example shows how to add the ability to query layers through the use of HTML templates.



Select the drop-down menu on the left, and select "nquery", and then click on the map; a query of all layers is executed.


MapServer Workshop - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://127.0.0.1/cgi-bin/mapserv.exe?map=C%3A%5COSGeo4W%2f
Google
MapServer Workshop
Done

レイヤ: 道路

	状態	有料	種別	国道番号	幅員	橋	トンネル	雪覆い
1	供用中	無料	一般道	0	3.0m以上5.5m未満			
2	供用中	無料	一般道	0	3.0m以上5.5m未満			
3	供用中	無料	一般道	0	3.0m以上5.5m未満			
4	供用中	無料	一般道	0	3.0m以上5.5m未満			
5	供用中	無料	一般道	0	3.0m以上5.5m未満			
6	供用中	無料	一般道	0	3.0m以上5.5m未満			
7	供用中	無料	一般道	0	3.0m以上5.5m未満			
8	供用中	無料	一般道	0	3.0m以上5.5m未満			
9	供用中	無料	一般道	0	3.0m以上5.5m未満			

– 戻る (back) –

A window should open displaying the query results.

The template file used in this exercise follows:

```
<!-- MapServer Template -->
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
<title>Æ‰ï</title>
<script>
  function doSubmit()
  {
    document.mapform.submit();
  }
</script>
</head>
<body BGCOLOR="white" TEXT="#000000" LINK="#0000FF" VLINK="#0000FF">
```

```
<table width="601" align="center" border="0">
<tr>
 <td width="593" align=center>
  <table cellpadding=10 >
  <tr>
    <td align=left><img src="/mapserver-tutorial/img/powered_mapserver.jpg"></td>
    <td align=left><h3>Æ‰ï (adding layer queries)</h3></td>
  </tr>
  </table>
  <br>
<!-- START OF MAPSERVER FORM -->
<form name="mapform" method="GET" action="/cgi-bin/mapserv.exe">
<!-- HIDDEN MAPSERVER CGI VARIABLES -->
<input type="hidden" name="map" value="[map]">
<input type="hidden" name="imgext" value="[mapext]">
<input type="hidden" name="mapext" value="[mapext]">
  <tr>
    <td>
       <table width="400" border="0">
      <tr>
        <td bgcolor="#425b7a" width="40%" >

            <!-- SPECIFY MAP MODE -->
           <div align="center"><font color=white>ƒ‚[ƒh: </font>
           <select name="mode">
              <option value="browse">•\Ž¦ (browse)</option>
              <option value="map">’n} (map)</option>
              <option value="query">’PˆêÆ‰ï (query)</option>
              <option value="nquery">•¡”Æ‰ï (nquery)</option>
             </select>
           </div>
         </td>
         <td bgcolor="#425b7a" width="15%">
           <!-- FORM SUBMIT BUTTON →
           <div align="center">
             <input type="button" name="submitter" value="ƒŠƒZƒbƒg (submit)"
onclick="doSubmit()" >
           </div>
         </td>
         <td bgcolor="#425b7a" width="45%">
              <!-- ZOOM/PAN CONTROLS -->
            <div align="center"><font color=white>ˆÚ“®/Šg‘å/k¬:</font>
                <select name="zoom">
                  <option value="4" [zoom_4_select]>Šg‘å 4x (zoom-in)</option>
                  <option value="3" [zoom_3_select]>Šg‘å 3x (zoom-in)</option>
                  <option value="2" [zoom_2_select]>Šg‘å 2x (zoom-in)</option>
                  <option value="1" [zoom_1_select]>ˆÚ“® (pan)</option>
                  <option value="-2" [zoom_-2_select]>k¬ 2x (zoom-out)</option>
                  <option value="-3" [zoom_-3_select]>k¬ 3x (zoom-out)</option>
                  <option value="-4" [zoom_-4_select]>k¬ 4x (zoom-out)</option>
                </select>
              </div>
         </td>
      </tr>
      <tr>
        <!-- DISPLAY THE MAPSERVER-CREATED MAP IMAGE -->
         <td colspan="3" align="right" valign="top">
            <input type="image" name="img" src="[img]"
                           width="550" height="450" border="1">
         </td>
           <td valign=top >
            <!-- DISPLAY THE LEGEND IMAGE -->
            <img src="[legend]">
           </td>
      </tr>
       </table>
    </td>
  </tr>
  <tr>
```

```
    <td>
       <table width=560 border="0">
         <tr>
           <td bgcolor="#425b7a">
             <font color="white">
             <div align="center">
             <input type=checkbox name=layer value="s" [s_check]
onclick="doSubmit()" >s
             <input type=checkbox name=layer value="�much" ["S"¹_check]
onclick="doSubmit()" >"S"¹
             <input type=checkbox name=layer value="‰Íì" [‰Íì_check]
onclick="doSubmit()" >‰Íì
             <input type=checkbox name=layer value="…ˆæ" […ˆæ_check]
onclick="doSubmit()" >…ˆæ
             </font>
           </td>
         </tr>
         <tr>
           <td align=right>
              <!-- DISPLAY THE SCALEBAR IMAGE -->
              <img src="[scalebar]" border="0">
           </td>
         </tr>
       </table>
    </td>
  </tr>
</form>
</table>
<br>
<hr noshade width="80%" size="1" align="center">
<center>
<a href="/mapserver-tutorial/example11/example_map.html">-ƒ}ƒbƒvƒtƒ@ƒCƒ‹  (mapfile)
-</a>&nbsp;&nbsp;
<a href="/mapserver-tutorial/index.html">- –ß‚é (back) -</a>
</center>
</body>
</html>
```

Notice the MODE dropdown and its values; MapServer uses the “query” mode to query a single layer, and the “nquery” mode to query multiple layers.

The mapfile used in this exercise is:

```
MAP
  IMAGETYPE   PNG
  EXTENT      139.72520 35.67139 139.78845 35.70731
  SIZE        550 450
  IMAGECOLOR  255 255 255
  SHAPEPATH   "../data"
  FONTSET     ../fonts/fonts.txt
  UNITS dd
  WEB
    TEMPLATE  'example_template.html'
    IMAGEPATH "@osgeo4w@/tmp/ms_tmp/"
    IMAGEURL "/ms_tmp/"
    MINSCALEDENOM  1000
    MAXSCALEDENOM  70000
  END
  SCALEBAR
    UNITS kilometers
    BACKGROUNDCOLOR 250 150 150
    COLOR 0 0 0
    TRANSPARENT on
    STYLE 0
    STATUS on
    LABEL
      COLOR 0 0 0
      SIZE tiny
    END
  END
  LEGEND
    KEYSIZE 40 20
    KEYSPACING 10 10
    OUTLINECOLOR 0 0 0
    IMAGECOLOR 255 255 255
    LABEL
      TYPE TRUETYPE
      FONT pmincho
      COLOR 0 0 0
      SIZE 12
      POSITION CL
      PARTIALS FALSE
      BUFFER 3
      ENCODING "SHIFT_JIS"
    END
    STATUS ON
  END
  LAYER
    NAME ""¹~H"
    DATA dourokukan
    STATUS DEFAULT
    TYPE LINE
    MAXSCALEDENOM 50000
    TOLERANCE 4
    TOLERANCEUNITS pixels
    HEADER   dourokukan_header.html
    TEMPLATE dourokukan_template.html
    FOOTER   dourokukan_footer.html
    CLASS
      NAME ""¹~H"
      STYLE
        COLOR 187 187 127
      END
    END
  END
  LAYER
    NAME "s"
    DATA gyouseikai
    STATUS ON
    TYPE LINE
    CLASS
      NAME "s"
      STYLE
        COLOR 227 127 227
      END
    END
  END
  LAYER
    NAME "“S”¹"
    DATA tetsudokukan
    STATUS ON
    TYPE LINE
    CLASS
      NAME "“S”¹"
      STYLE
        COLOR 128 128 128
      END
    END
  END
  LAYER
    NAME "‰Íì"
    DATA kasenkukan
    STATUS ON
    TYPE LINE
    CLASS
      NAME "‰Íì"
      STYLE
        COLOR 0 255 255
      END
    END
  END
  LAYER
    NAME "…ˆæ"

    DATA suiikikai

    STATUS ON

    TYPE LINE

    CLASS

      NAME "…ˆæ"

      STYLE

        COLOR 0 0 255

      END

    END

  END

  LAYER

    NAME "’n–¼"

    DATA chimei

   STATUS DEFAULT

    TYPE POINT

    LABELITEM "NAMAE"

    CLASS

      NAME "’n–¼"

      STYLE

        COLOR 10 100 100

      END

      LABEL

        TYPE TRUETYPE

        FONT pgothic

        COLOR 220 20 20

        SIZE 7

        POSITION CL

        PARTIALS FALSE

        BUFFER 5

        ENCODING "SHIFT_JIS"

      END

    END

  END
END
```

Notice the TEMPLATE parameter set in *dourokukan* layer; the new parameters are described below:

TOLERANCE
This is the range of acceptance for a given query.

TOLERANCEUNITS
Specify the units for the TOLERANCE parameter.

HEADER
The header file displays the HTML shown before the results.

TEMPLATE
The template file itself contains the query results.

FOOTER
The footer file contains the HTML shown after the query results.

dourokukan_header.html

```
<!-- MapServer Template -->
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
<title>MapServer Workshop</title>
</head>
<font size+3><b>¥ì¥¤¥ä: Æ»Ï©</b></font><p>
<hr noshade width="80%" size="1" align="center">
<br>
<div align=center>
<table cellpadding=5 cellspacing=2 border=1>
<tr bgcolor=#CCCCCC>
<td bgcolor=#ffffff>&nbsp;</td>
<th>¾õÂÖ</th>
<th>ÍÎÁ</th>
<th>¼ïÊÌ</th>
<th>¹ñÆ»ÈÖ¹æ</th>
<th>Éý°÷</th>
<th>¶¶</th>
<th>¥È¥ó¥Í¥ë</th><th>ÀãÊ¤¤¤</th></tr>
```

dourokukan_template.html

```
<!-- MapServer Template -->
<tr>
  <td>[JOUTAI]</td>
  <td>[YURYO]</td>
  <td>[SYUBETSU]</td>
  <td>[KOKUBANGO]</td>
  <td>[FUKUIN]</td>
  <td>[HASHI]</td>
  <td>[TONNERU]</td>
  <td>[YUKIOOI]</td>
</tr>
```

dourorkukan_footer.html

```
<!-- MapServer Template -->
</table>
</div>
<br>
<hr noshade width="80%" size="1" align="center">
<div align="center"><b><a href="javascript:history.back()">- Ìá¤ë (back) -</a><br>
</html>
```

Note that each template file (HEADER, FOOTER, TEMPLATE) must always contain the MapServer "magic string" of <!-- MapServer Template -->

The TEMPLATE contains field names in square brackets (such as "JOUTAI") that are populated with values by MapServer during run-time.

Using WMS Services with MapServer

About WMS

The Open Geospatial Consortium (OGC, http://www.opengeospatial.org/) is an international group of organizations that focus on sharing geospatial information through standards. One of the OGC's most widely used standards is the Web Map Server (WMS) specification, which shares images (such as GIF, PNG, JPEG) of spatial data across remote servers (in other words: the actual data is not transferred, only an image of the data is transferred from a remote WMS server).

MapServer has always been one of the early adopters of OGC standards. MapServer's OGC standard support is well documented at: http://www.mapserver.org/ogc/ This section will discuss MapServer's WMS support. MapServer can act both as a WMS server (serving images of data to other servers), or as a WMS client (request and use remote images of data from other servers). Some relevant documents from MapServer are:

- MapServer WMS Server Howto: http://www.mapserver.org/ogc/wms_server.html
- MapServer WMS Client Howto: http://www.mapserver.org/ogc/wms_client.html

Exercise 12: Serving WMS Layers Through MapServer (WMS Server)

In this exercise we will add a few mandatory parameters to our mapfile so that it can be served through the WMS standard.

Step1: Configure Your Mapfile

The mapfile used in this exercise is:

```
MAP
  NAME         "mapserver_tutorial_wms_server"
  IMAGETYPE    PNG
  EXTENT       139.72520 35.67139 139.78845 35.70731
  SIZE         550 450
  IMAGECOLOR   255 255 255
  SHAPEPATH    "../data"
  FONTSET      ../fonts/fonts.txt
  UNITS dd
  WEB
    TEMPLATE  'example_template.html'
    IMAGEPATH "@osgeo4w@/tmp/ms_tmp/"
    IMAGEURL "/ms_tmp/"
    MINSCALEDENOM  1000
    MAXSCALEDENOM  70000
    METADATA
      "wms_title"           "MapServer Tutorial WMS Server"
      "wms_onlineresource" "http://127.0.0.1/cgi-bin/mapserv.exe?
MAP=@osgeo4w@/apps/mapserver-tutorial/example12.map"
      "wms_srs"             "EPSG:4301 EPSG:4269 EPSG:4326"
      "wms_abstract"      "This demonstration server shows how to setup a
                           MapServer .map file to serve data through the WMS
                           standard."
    END
  END
  PROJECTION
    "init=epsg:4301"
  END
  LAYER
    NAME "roads"
    METADATA
    "wms_title"    "Roads"
    END
    DATA dourokukan
    STATUS ON
    TYPE LINE
    MAXSCALEDENOM 50000
    CLASS
      NAME "roads"
      STYLE
        COLOR 187 187 127
      END
    END
    PROJECTION
      "init=epsg:4301"
    END
  END
  LAYER
    NAME "administrative"
    METADATA
    "wms_title"    "Administrative"
    END
    DATA gyouseikai
    STATUS ON
    TYPE LINE
    CLASS
      NAME "Administrative"
      STYLE
        COLOR 227 127 227
      END
    END
    PROJECTION
      "init=epsg:4301"
    END
  END
  LAYER
    NAME "rail"
    METADATA
    "wms_title"    "Rail"
    END
    DATA tetsudokukan
    STATUS ON
    TYPE LINE
    CLASS
      NAME "Rail"
      STYLE
        COLOR 128 128 128
      END
    END
    PROJECTION
      "init=epsg:4301"
    END
  END
  LAYER
    NAME "rivers"
    METADATA
    "wms_title"    "Rivers"
    END
    DATA kasenkukan
    STATUS ON
    TYPE LINE
    CLASS
      NAME "Rivers"
      STYLE
        COLOR 0 255 255
      END
    END
    PROJECTION
      "init=epsg:4301"
    END
  END
  LAYER
    NAME "water"
    METADATA
    "wms_title"    "Water"
    END
    DATA suiikikai
    STATUS ON
    TYPE LINE
    CLASS
      NAME "Water"
      STYLE
        COLOR 0 0 255
      END
    END
    PROJECTION
      "init=epsg:4301"
    END
  END
  LAYER
    NAME "places"
    METADATA
    "wms_title"    "Place name"
    END
    DATA chimei
    STATUS ON
    TYPE POINT
    LABELITEM "NAMAE"
    CLASS
      NAME "Places"
      STYLE
        COLOR 10 100 100
      END
      LABEL
        TYPE TRUETYPE
        FONT pgothic
        COLOR 220 20 20
        SIZE 7
        POSITION CL
        PARTIALS FALSE
        BUFFER 5
        ENCODING "SHIFT_JIS"
      END
    END
    PROJECTION
      "init=epsg:4301"
    END
  END END
```

The highlighed text shows the required parameters to serve data through the WMS standard with MapServer, and they are described next:

WMS_TITLE

This metadata must be set in the WEB object and in every layer that you want served through WMS. Use it to describe your data. All possible WMS metadata items are documented at: http://www.mapserver.org/ogc/wms_server.html#reference-section

WMS_ONLINERESOURCE

This metadata item must be set in the WEB object; it is the URL to your mapserv.exe executable, and in our example includes the full path to our mapfile. More information about this metadata item can be read at: http://www.mapserver.org/ogc/wms_server.html#more-about-the-online-resource-url

PROJECTION

In order to serve data through any OGC standard, you must specify the projection that the data is in. Each layer must have a PROJECTION object, as well as a PROJECTION set at the MAP object level. Please read the MapServer documentation on the PROJECTION object: http://www.mapserver.org/mapfile/projection.html

Step2: Validate Your Server's Capabilities

Now that we have configured our mapfile for serving WMS, we must check that it is configured properly for the standard. We do this by using the onlineresource URL in a browser, and add the following to it: "&SERVICE=WMS&VERSION=1.1.1&REQUEST=GetCapabilities ". In our example, the full request is:

```
http://127.0.0.1/cgi-bin/mapserv.exe?MAP=C:/OSGeo4W/apps/mapserver-
tutorial/example12/example.map&SERVICE=wms&VERSION=1.1.1&REQUEST=GetCapabilities
```

Generally that request is referred to as the GetCapabilities request. If you execute it in a browser, you will be asked to save a file: rename this file so that it has a .xml extension (such as 'mapserv.xml'). Then open this file in a text editor. You should see a response from your WMS server, describing the layers available, such as:

```
<?xml version='1.0' encoding="ISO-8859-1" standalone="no" ?>
<!DOCTYPE WMT_MS_Capabilities SYSTEM
"http://schemas.opengis.net/wms/1.1.1/WMS_MS_Capabilities.dtd"
 [
 <!ELEMENT VendorSpecificCapabilities EMPTY>
 ]>  <!-- end of DOCTYPE declaration -->
<WMT_MS_Capabilities version="1.1.1">
<!-- MapServer version 5.2.2 OUTPUT=GIF OUTPUT=PNG OUTPUT=JPEG OUTPUT=WBMP
OUTPUT=SWF OUTPUT=SVG SUPPORTS=PROJ SUPPORTS=AGG SUPPORTS=FREETYPE SUPPORTS=ICONV
SUPPORTS=FRIBIDI SUPPORTS=WMS_SERVER SUPPORTS=WMS_CLIENT SUPPORTS=WFS_SERVER
SUPPORTS=WFS_CLIENT SUPPORTS=WCS_SERVER SUPPORTS=SOS_SERVER SUPPORTS=FASTCGI
SUPPORTS=THREADS SUPPORTS=GEOS SUPPORTS=RGBA_PNG INPUT=JPEG INPUT=POSTGIS INPUT=OGR
INPUT=GDAL INPUT=SHAPEFILE -->
<Service>
  <Name>OGC:WMS</Name>
  <Title>MapServer Tutorial WMS Server</Title>
  <Abstract>This demonstration server shows how to setup a MapServer .map file to
serve data through the WMS standard.</Abstract>
  <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
xlink:href="http://127.0.0.1/cgi-bin/mapserv.exe?MAP=C:\OSGeo4W/apps/mapserver-
tutorial/example12.map&amp;"/>
  <ContactInformation>
  </ContactInformation>
</Service>
<Capability>
  <Request>
    <GetCapabilities>
      <Format>application/vnd.ogc.wms_xml</Format>
      <DCPType>
        <HTTP>
          <Get><OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
xlink:href="http://127.0.0.1/cgi-bin/mapserv.exe?MAP=C:\OSGeo4W/apps/mapserver-
tutorial/example12.map&amp;"/></Get>
          <Post><OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
xlink:href="http://127.0.0.1/cgi-bin/mapserv.exe?MAP=C:\OSGeo4W/apps/mapserver-
tutorial/example12.map&amp;"/></Post>
        </HTTP>
      </DCPType>
    </GetCapabilities>
    <GetMap>
      <Format>image/gif</Format>
      <Format>image/png</Format>
      <Format>image/png; mode=24bit</Format>
      <Format>image/jpeg</Format>
      <Format>image/vnd.wap.wbmp</Format>
      <Format>image/tiff</Format>
      <Format>image/svg+xml</Format>
      <DCPType>
        <HTTP>
          <Get><OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
xlink:href="http://127.0.0.1/cgi-bin/mapserv.exe?MAP=C:\OSGeo4W/apps/mapserver-
tutorial/example12.map&amp;"/></Get>
          <Post><OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
xlink:href="http://127.0.0.1/cgi-bin/mapserv.exe?MAP=C:\OSGeo4W/apps/mapserver-
tutorial/example12.map&amp;"/></Post>
        </HTTP>
      </DCPType>
    </GetMap>
    <GetFeatureInfo>
      <Format>text/plain</Format>
      <Format>application/vnd.ogc.gml</Format>
      <DCPType>
        <HTTP>
          <Get><OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
xlink:href="http://127.0.0.1/cgi-bin/mapserv.exe?MAP=C:\OSGeo4W/apps/mapserver-
tutorial/example12.map&amp;"/></Get>
          <Post><OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
xlink:href="http://127.0.0.1/cgi-bin/mapserv.exe?MAP=C:\OSGeo4W/apps/mapserver-
tutorial/example12.map&amp;"/></Post>
        </HTTP>
      </DCPType>
    </GetFeatureInfo>
    <DescribeLayer>
      <Format>text/xml</Format>
      <DCPType>
```

```
        <HTTP>
          <Get><OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
xlink:href="http://127.0.0.1/cgi-bin/mapserv.exe?MAP=C:\OSGeo4W/apps/mapserver-
tutorial/example12.map&amp;"/></Get>
          <Post><OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
xlink:href="http://127.0.0.1/cgi-bin/mapserv.exe?MAP=C:\OSGeo4W/apps/mapserver-
tutorial/example12.map&amp;"/></Post>
        </HTTP>
      </DCPType>
    </DescribeLayer>
    <GetLegendGraphic>
      <Format>image/gif</Format>
      <Format>image/png</Format>
      <Format>image/png; mode=24bit</Format>
      <Format>image/jpeg</Format>
      <Format>image/vnd.wap.wbmp</Format>
      <DCPType>
        <HTTP>
          <Get><OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
xlink:href="http://127.0.0.1/cgi-bin/mapserv.exe?MAP=C:\OSGeo4W/apps/mapserver-
tutorial/example12.map&amp;"/></Get>
          <Post><OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
xlink:href="http://127.0.0.1/cgi-bin/mapserv.exe?MAP=C:\OSGeo4W/apps/mapserver-
tutorial/example12.map&amp;"/></Post>
        </HTTP>
      </DCPType>
    </GetLegendGraphic>
    <GetStyles>
      <Format>text/xml</Format>
      <DCPType>
        <HTTP>
          <Get><OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
xlink:href="http://127.0.0.1/cgi-bin/mapserv.exe?MAP=C:\OSGeo4W/apps/mapserver-
tutorial/example12.map&amp;"/></Get>
          <Post><OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
xlink:href="http://127.0.0.1/cgi-bin/mapserv.exe?MAP=C:\OSGeo4W/apps/mapserver-
tutorial/example12.map&amp;"/></Post>
        </HTTP>
      </DCPType>
    </GetStyles>
  </Request>
  <Exception>
    <Format>application/vnd.ogc.se_xml</Format>
    <Format>application/vnd.ogc.se_inimage</Format>
    <Format>application/vnd.ogc.se_blank</Format>
  </Exception>
  <VendorSpecificCapabilities />
  <UserDefinedSymbolization SupportSLD="1" UserLayer="0" UserStyle="1"
RemoteWFS="0"/>
  <Layer>
    <Name>mapserver_tutorial_wms_server</Name>
    <Title>MapServer Tutorial WMS Server</Title>
    <SRS>EPSG:4301</SRS>
    <SRS>EPSG:4269</SRS>
    <SRS>EPSG:4326</SRS>
    <LatLonBoundingBox minx="139.725" miny="35.6714" maxx="139.788"
maxy="35.7073" />
    <BoundingBox SRS="EPSG:4301"
                minx="139.725" miny="35.6714" maxx="139.788" maxy="35.7073" />
    <ScaleHint min="0.498902848429637" max="34.9231993900746" />
    <Layer queryable="0" opaque="0" cascaded="0">
        <Name>roads</Name>
        <Title>Roads</Title>
        <SRS>EPSG:4301</SRS>
        <LatLonBoundingBox minx="139.563" miny="35.5247" maxx="139.918"
maxy="35.8176" />
        <BoundingBox SRS="EPSG:4301"
                    minx="139.563" miny="35.5247" maxx="139.918" maxy="35.8176" />
        <Style>
          <Name>default</Name>
          <Title>default</Title>
          <LegendURL width="70" height="23">
             <Format>image/png</Format>
             <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
xlink:type="simple" xlink:href="http://127.0.0.1/cgi-bin/mapserv.exe?
```

```
MAP=C:\OSGeo4W/apps/mapserver-
tutorial/example12.map&amp;version=1.1.1&amp;service=WMS&amp;request=GetLegendGraph
ic&amp;layer=roads&amp;format=image/png&amp;STYLE=default"/>
          </LegendURL>
        </Style>
        <ScaleHint min="0" max="24.9451424214819" />
    </Layer>
    . . .
</Capability>
</WMT_MS_Capabilities>
```

If you are going to add a WMS layer from a remote server, it is recommended that you look at the server's GetCapabilities document. In our example above, you can see that our abstract and titles for all of the layers are displayed in the GetCapabilities document, as well as other important information such as what image formats the data is available in, and what projections (or "SRS") we can request the data in. If you have a problem with your WMS service, a "*WARNING*" tag will be included in your GetCapabilities document.

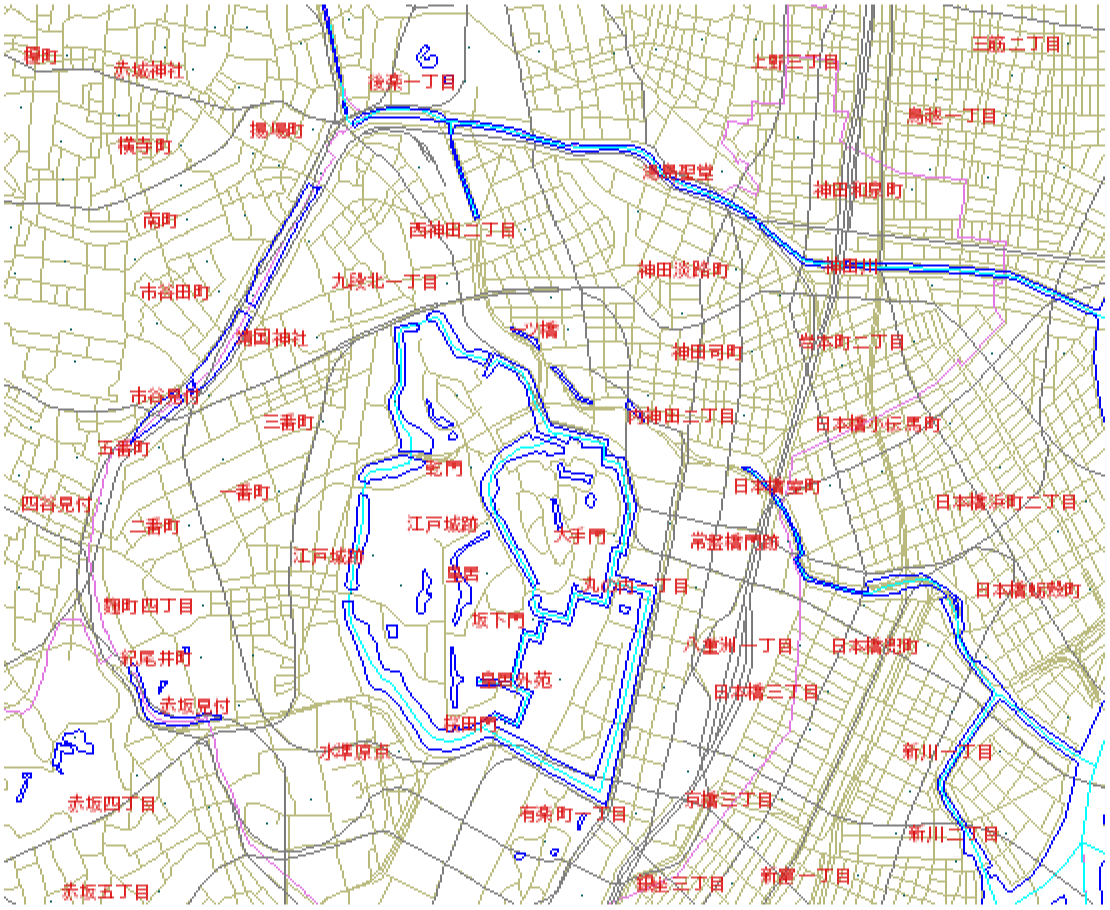
Step3: Request a Map from your Server

Now that you have verified that the capabilities of your WMS server are valid, you can now request an map image from your WMS server. The WMS specification uses a *GetMap* request to request a map from a remote WMS server. Please see the MapServer documentation for a definition of all of the parameters required for a GetMap request: http://www.mapserver.org/ogc/wms_server.html#test-with-a-getmap-request

In our example, our GetMap requests looks like this:

```
http://127.0.0.1/cgi-bin/mapserv.exe?MAP=C:\OSGeo4W/apps/mapserver-
tutorial/example12/example.map&SERVICE=wms&VERSION=1.1.1&REQUEST=GetMap&LAYERS=maps
erver_tutorial_wms_server&SRS=EPSG:4301&BBOX=139.72520,35.67139,139.78845,35.70731&
FORMAT=image/png&WIDTH=550&HEIGHT=450
```

In our example, the GetMap request will return an image such as:



MapServer will return a map image with given a GetMap request, or it will return an error.

Exercise 13: Consuming Remote WMS Layers in MapServer (WMS Client)

In this exercise we will add a layer from a remote WMS service to our mapfile.

Step1: Choose a remote WMS Service

Locating WMS services is not easy, as there is no official list of services (at the time of writing this document). For your benefit here are some possible WMS services:

Description	More Information	Onlineresource URL
Base map data for Japan	http://www.finds.jp/wsdocs/kibanwms/index.html	http://www.finds.jp/ws/kiban25000wms.cgi?
Landslide Map Database for Japan	http://lsweb1.ess.bosai.go.jp/jisuberi/jisuberi_mini/index.asp	http://lsweb1.ess.bosai.go.jp/wmsconnector/com.esri.wms.Esrimap
OnEarth, JPL WMS Server	http://onearth.jpl.nasa.gov/	http://wms.jpl.nasa.gov/wms.cgi
Geography Network for Japan	http://www.geographynetwork.ne.jp/	http://www.geographynetwork.ne.jp/ogc/wms?ServiceName=basemap_wms
Agricultural/Environmental for Japan	http://habs.dc.affrc.go.jp/index.html	http://habs.dc.affrc.go.jp/geowebcache/wms?
GeoGrid WMS	http://docs.geogrid.org/Applications/PublicWMS	http://carteb.geogrid.org/mapserv/qqm

Step2: Review the WMS Server's GetCapabilities

In this example we chose GeoGrid's geology WMS service (http://geodata1.geogrid.org/mapserv/g1000k/g1000ke?). To look at their service's capabilities we would execute a GetCapabilities request, such as:

http://geodata1.geogrid.org/mapserv/g1000k/g1000ke?SERVICE=wms&VERSION=1.1.1&REQUEST=GetCapabilities

A portion of their GetCapabilities response follows:

```
 <Layer>
    <Name>g1000ke</Name>
    <Title>1/1,000,000 geological map of Japan (English)</Title>
    <SRS>esri:54004</SRS>
    <SRS>epsg:4326</SRS>
    <SRS>epsg:4612</SRS>
    <SRS>epsg:4301</SRS>
    <LatLonBoundingBox minx="122.943" miny="20.4586" maxx="153.99"
maxy="45.5565" />
    <BoundingBox SRS="EPSG:4612"
                minx="122.943" miny="20.4586" maxx="153.99" maxy="45.5565" />
    <Layer queryable="0" opaque="0" cascaded="0">
        <Name>area</Name>
        <Title>Area</Title>
        <Abstract>1/1,000,000 geological map (Polygon data)</Abstract>
        <SRS>EPSG:4612</SRS>
        <LatLonBoundingBox minx="122.943" miny="20.4586" maxx="153.99"
maxy="45.5565" />
        <BoundingBox SRS="EPSG:4612"
                    minx="122.943" miny="20.4586" maxx="153.99" maxy="45.5565" />
        <Style>
          <Name>default</Name>
          <Title>default</Title>
          <LegendURL width="20" height="15">
             <Format>image/png</Format>
             <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
xlink:type="simple" xlink:href="http://geodata1.geogrid.org/mapserv/g1000k/g1000ke?
version=1.1.1&amp;service=WMS&amp;request=GetLegendGraphic&amp;layer=area&amp;forma
t=image/png"/>
          </LegendURL>
        </Style>
        <ScaleHint min="0" max="4989.02848429637" />
    </Layer>
    <Layer queryable="0" opaque="0" cascaded="0">
        <Name>line</Name>
        <Title>Line</Title>
        <Abstract>1/1,000,000 geological map (Fault and boundary data)</Abstract>
        <SRS>EPSG:4612</SRS>
        <LatLonBoundingBox minx="122.943" miny="20.4586" maxx="153.99"
maxy="45.5565" />
        <BoundingBox SRS="EPSG:4612"
                    minx="122.943" miny="20.4586" maxx="153.99" maxy="45.5565" />
        <Style>
          <Name>default</Name>
          <Title>default</Title>
          <LegendURL width="20" height="15">
             <Format>image/png</Format>
             <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink"
xlink:type="simple" xlink:href="http://geodata1.geogrid.org/mapserv/g1000k/g1000ke?
version=1.1.1&amp;service=WMS&amp;request=GetLegendGraphic&amp;layer=line&amp;forma
t=image/png"/>
          </LegendURL>
        </Style>
        <ScaleHint min="0" max="4989.02848429637" />
    </Layer>
    <Layer queryable="0" opaque="0" cascaded="0">
        <Name>label</Name>
        <Title>Label</Title>
        <Abstract>1/1,000,000 geological map (Label information)</Abstract>
        <SRS>EPSG:4612</SRS>
        <LatLonBoundingBox minx="122.943" miny="20.4586" maxx="153.99"
maxy="45.5565" />
        <BoundingBox SRS="EPSG:4612"
                    minx="122.943" miny="20.4586" maxx="153.99" maxy="45.5565" />
        <ScaleHint min="0" max="4989.02848429637" />
    </Layer>
```

From the GetCapabilities response, we can see that there is a “ g1000ke” layer available, and that server is providing the data in the same projection as our local data (EPSG:4301).

Step3: Add a WMS Layer to our Mapfile

To add a WMS layer to our mapfile, you can follow the instructions in the MapServer WMS Client document: http://www.mapserver.org/ogc/wms_client.html

Our example WMS layer follows:

```
LAYER
  NAME "geology-wms"
  TYPE RASTER
  STATUS OFF
  CONNECTION "http://geodata1.geogrid.org/mapserv/g1000k/g1000ke?"
  CONNECTIONTYPE WMS
  METADATA
    "wms_srs"              "EPSG:4301"
    "wms_name"             "g1000ke"
    "wms_server_version"   "1.1.1"
    "wms_format"           "image/png"
  END
END
```

Here is a description of the important parameters of this layer:

TYPE

Since we are requesting a PNG image from the WMS server, the type is set to RASTER.

CONNECTION

This is the onlineresource URL for the remote WMS server.

CONNECTIONTYPE

Must be set to 'WMS'.

WMS_SRS

This is the projection that we request the data to be in. We have already verified that their remote WMS server serves the data in this projection.

WMS_NAME

This is the name of the remote WMS layer, as specified in their GetCapabilities response.

WMS_SERVER_VERSION

This is the WMS version as obtained through the GetCapabilities response.

WMS_FORMAT

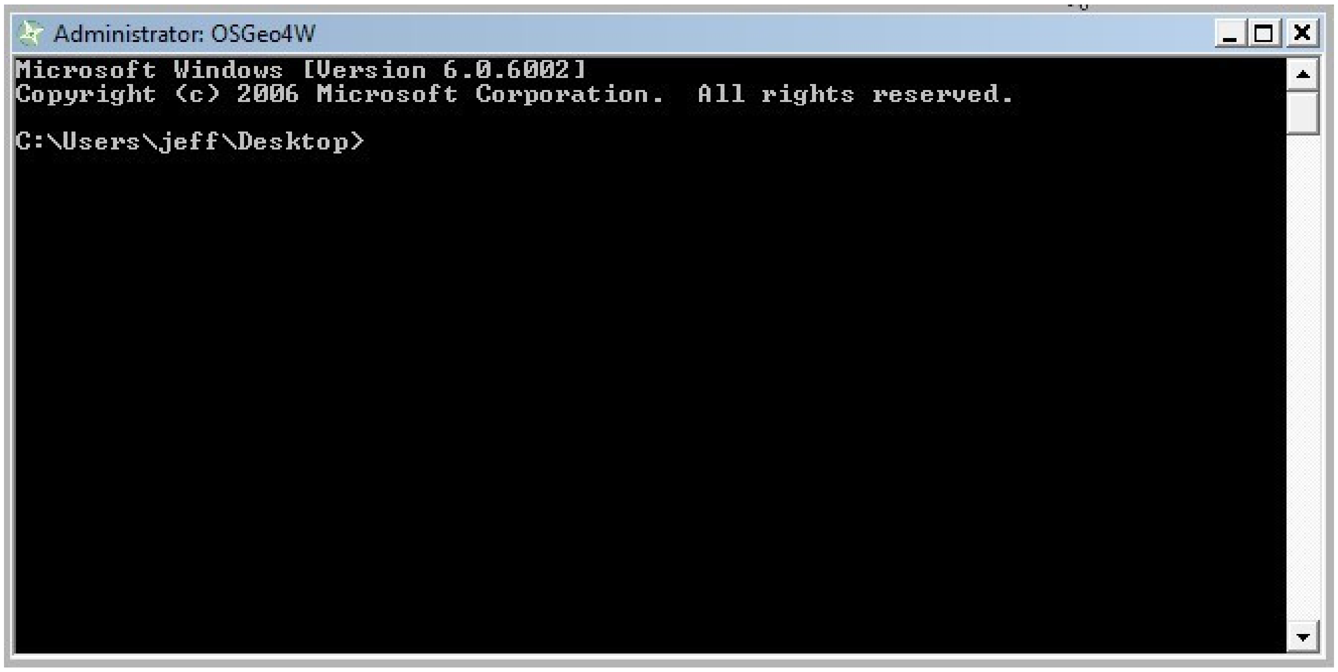
This is the format that we want the map image in. We have already verified that the PNG format is available on their WMS server.

Step4: Generate a Map

Finally we can try to view our local layers with this remote WMS layer. The exercise HTML page has a working link to generate a map image, but you can also use MapServer's shp2img.exe commandline utility to test your mapfile. More information about shp2img can be found at: http://www.mapserver.org/utilities/shp2img.html

Execute the following steps:

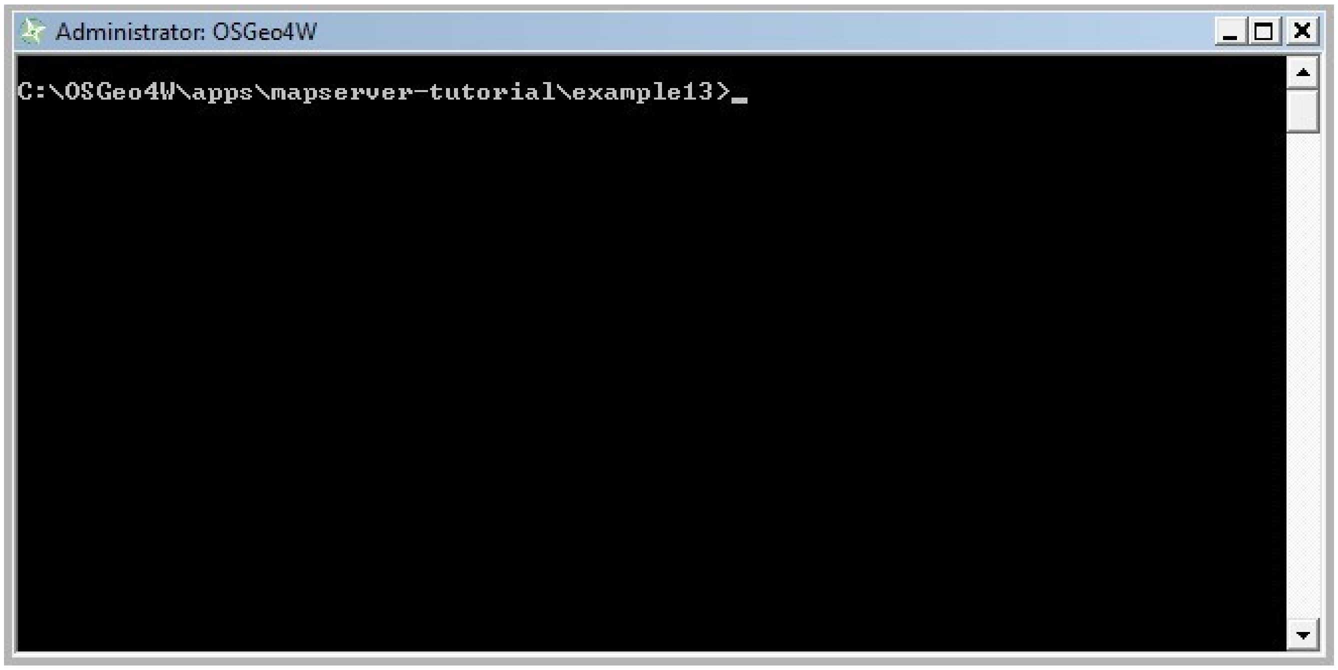
1) Click on the "OSGeo4W" shortcut on your desktop. A commandline shell window should open.



```
Microsoft Windows [Version 6.0.6002]
Copyright (c) 2006 Microsoft Corporation.  All rights reserved.

C:\Users\jeff\Desktop>
```

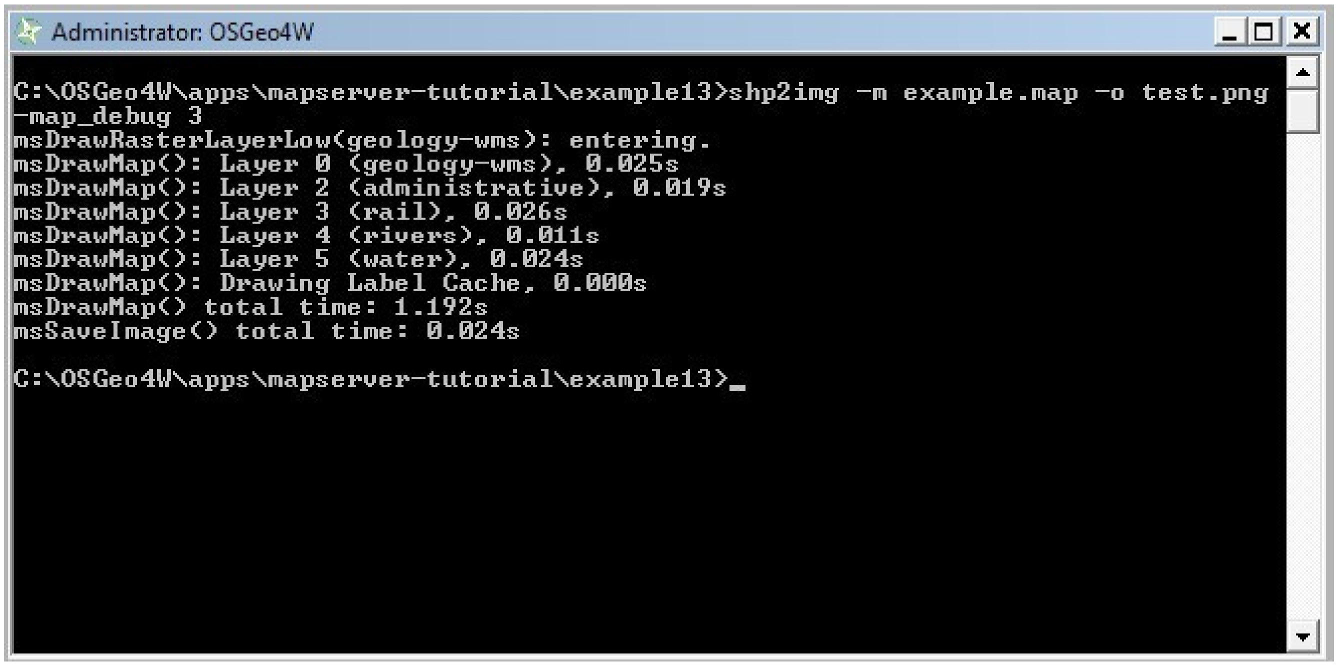
2) CD (change directory) to the example13 directory.



Administrator: OSGeo4W

```
C:\OSGeo4W\apps\mapserver-tutorial\example13>_
```

3) Execute: shp2img -m example.map -o test.png -map_debug 3
(this should generate a 'test.png' map image, or display an error)



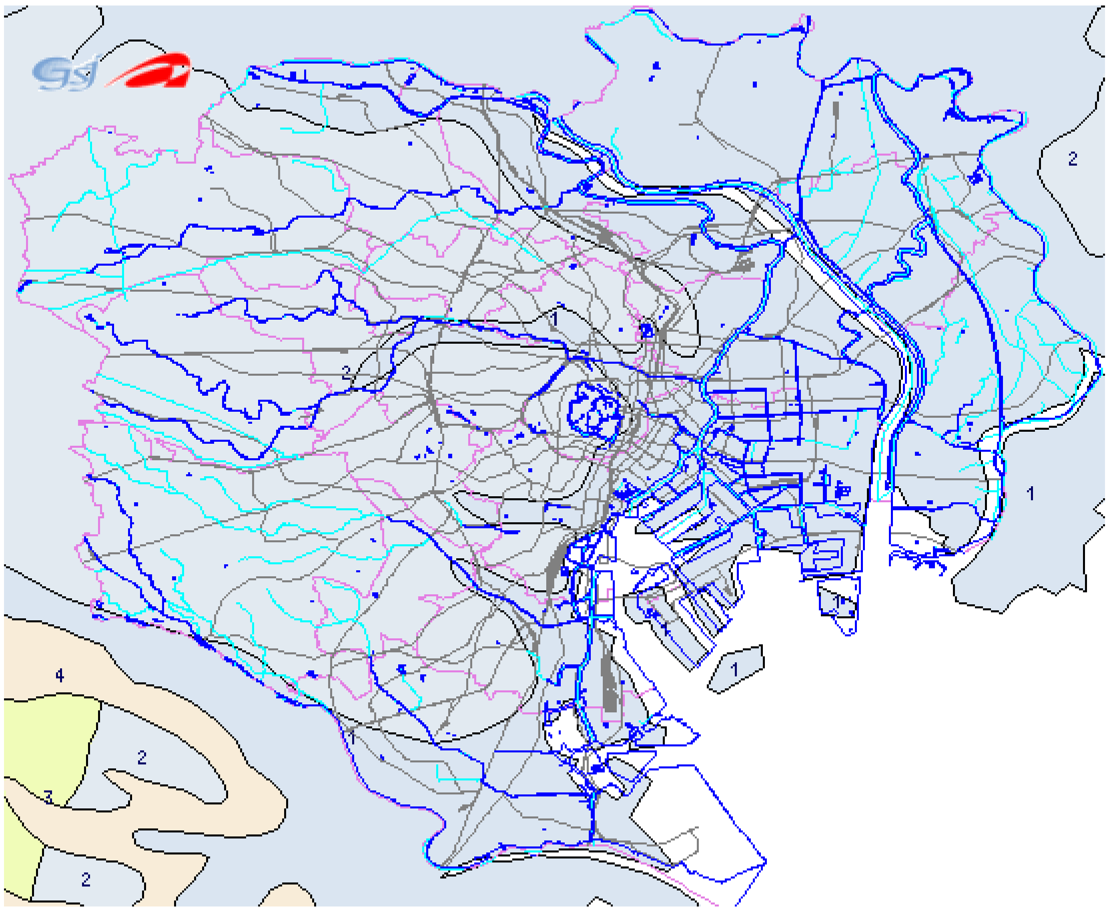
Administrator: OSGeo4W

```
C:\OSGeo4W\apps\mapserver-tutorial\example13>shp2img -m example.map -o test.png
-map_debug 3
msDrawRasterLayerLow(geology-wms): entering.
msDrawMap(): Layer 0 (geology-wms), 0.025s
msDrawMap(): Layer 2 (administrative), 0.019s
msDrawMap(): Layer 3 (rail), 0.026s
msDrawMap(): Layer 4 (rivers), 0.011s
msDrawMap(): Layer 5 (water), 0.024s
msDrawMap(): Drawing Label Cache, 0.000s
msDrawMap() total time: 1.192s
msSaveImage() total time: 0.024s

C:\OSGeo4W\apps\mapserver-tutorial\example13>_
```

4) Drag 'test.png' into your Web browser.

   (if you have a blank image, open example13/example.map in a text editor and replace all 'STATUS OFF' values with 'STATUS ON' and retry shp2img)



You should now see a map image of our local layers overlayed on top of the remote WMS server (geology) layer.

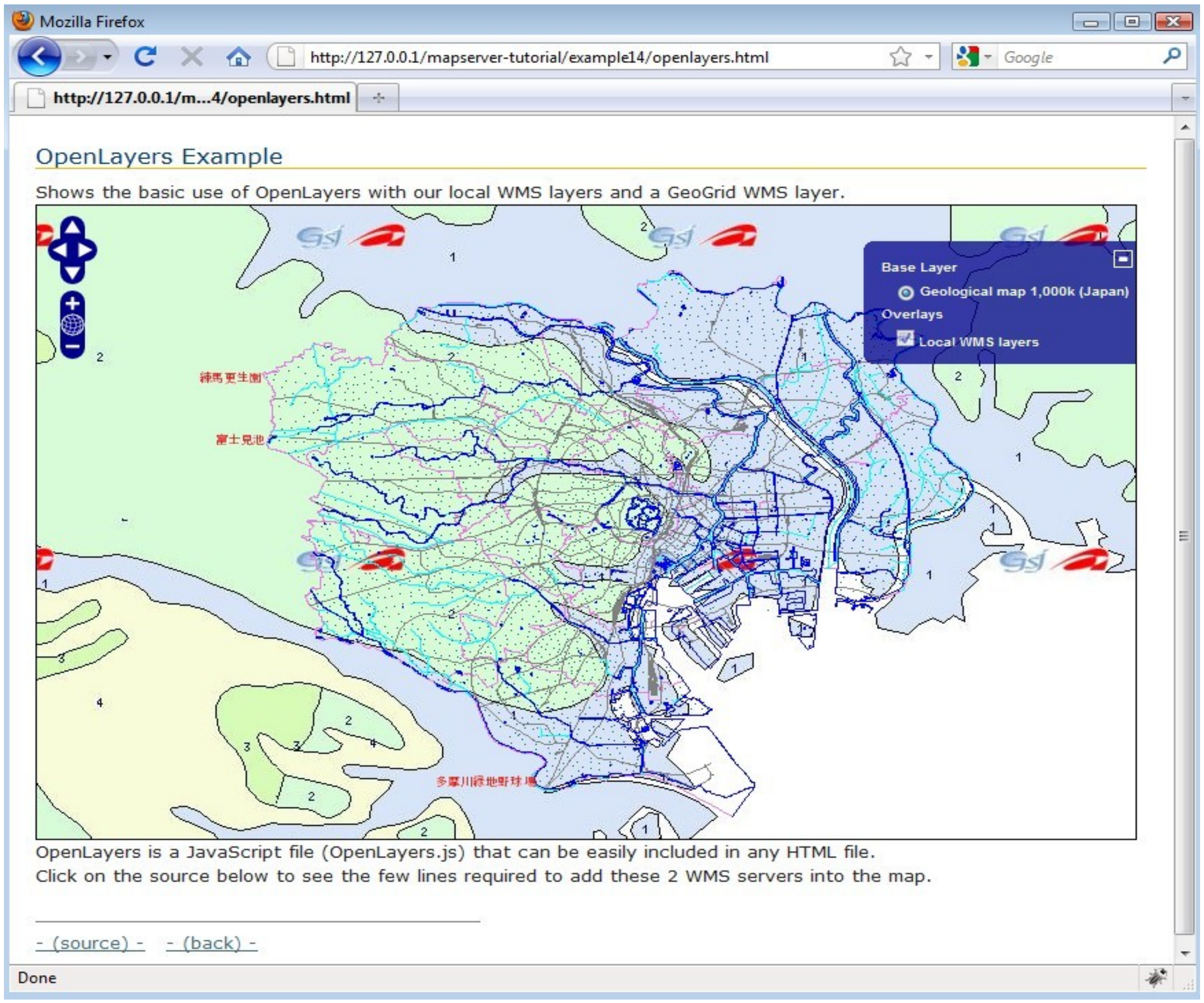
Using OpenLayers with Your MapServer Data

About OpenLayers

OpenLayers (http://www.openlayers.org/) is another project of the Open Source Geospatial Foundation (OSGeo), which is a very popular JavaScript API for creating online mapping applications. OpenLayers is designed for use with WMS layers, so we will use it in our example to display our WMS layers that we configured in Example12, and the GeoGrid WMS service.

Exercise 14: Creating a Simple OpenLayers Viewer

In this exercise we will use OpenLayers to create a simple JavaScript viewer for our own WMS-served layers, and the GeoGrid WMS layer.


Mozilla Firefox
http://127.0.0.1/mapserver-tutorial/example14/openlayers.html
Google
http://127.0.0.1/m...4/openlayers.html
Base Layer
Geological map 1,000k (Japan)
Overlays
Local WMS layers
- (source) -
- (back) -
Done

OpenLayers Example

Shows the basic use of OpenLayers with our local WMS layers and a GeoGrid WMS layer.

OpenLayers is a JavaScript file (OpenLayers.js) that can be easily included in any HTML file.
Click on the source below to see the few lines required to add these 2 WMS servers into the map.

The HTML file used to create this interface follows:

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1"/>
    <link rel="stylesheet" href="style.css" type="text/css" />
    <style type="text/css">
        #map {
         width: 800px;
            height: 500px;
            border: 1px solid black;
        }
    </style>
    <script src="OpenLayers.js"></script>
    <script type="text/javascript">
        var lon = 139.7245;
        var lat = 35.6816;
        var zoom = 11;
        var map, layer;
        function init(){
            map = new OpenLayers.Map( 'map' );

            g1000ke = new OpenLayers.Layer.WMS( "Geological map 1,000k (Japan)",
                     "http://geodata1.geogrid.org/mapserv/g1000k/g1000ke",
                     {layers: 'g1000ke'} );
            map.addLayer(g1000ke);

            localWMS = new OpenLayers.Layer.WMS( "Local WMS layers",
                            "http://127.0.0.1/cgi-bin/mapserv.exe?
MAP=C:/OSGeo4W/apps/mapserver-tutorial/example12/example.map",
                            {layers: 'mapserver_tutorial_wms_server', transparent:
true}, {isBaseLayer: false} );
            map.addLayer(localWMS);

            map.setCenter(new OpenLayers.LonLat(lon, lat), zoom);
            map.addControl( new OpenLayers.Control.LayerSwitcher() );
        }
    </script>
  </head>
  <body onload="init()">
    <h1 id="title">OpenLayers Example</h1>
    <div id="tags">
    </div>
    <p id="shortdesc">
        Shows the basic use of OpenLayers with our local WMS layers and a GeoGrid
WMS layer.
    </p>
    <div id="map" class="smallmap"></div>
    <div id="docs">
        OpenLayers is a JavaScript file (OpenLayers.js) that can be easily included
in any HTML file.<br>Click on the source
        below to see the few lines required to add these 2 WMS servers into the
map.<br>
          <br>
          <hr noshade width="40%" size="1" align="left">
          <a href="/mapserver-tutorial/example14/openlayers_source.html">- (source)
-</a>&nbsp;&nbsp;
          <a href="/mapserver-tutorial/index.html">- (back) -</a>
    </div>
  </body>
</html>
```

As you can see above, the HTML file required to generate this viewer is very small. The important parts of this file are as follows:

1) the <script> tag points to OpenLayers.js

   This is a compressed version of the OpenLayers library. Instructions to generate this file are found at: http://docs.openlayers.org/library/deploying.html

2) the OpenLayers.Layer.WMS constructor is used to point to our local WMS service, and the GeoGrid service. Documentation on the contructor is found at: http://dev.openlayers.org/apidocs/files/OpenLayers/Layer/WMS-js.html

3) the map.SetCenter function is used to specify the map extents in the viewer. Documentation on

this function can be found at: http://dev.openlayers.org/apidocs/files/OpenLayers/Map-js.html#OpenLayers.Map.setCenter

And that is all that is required to generate an OpenLayers viewer.

Using Quantum GIS with MapServer

About Quantum GIS

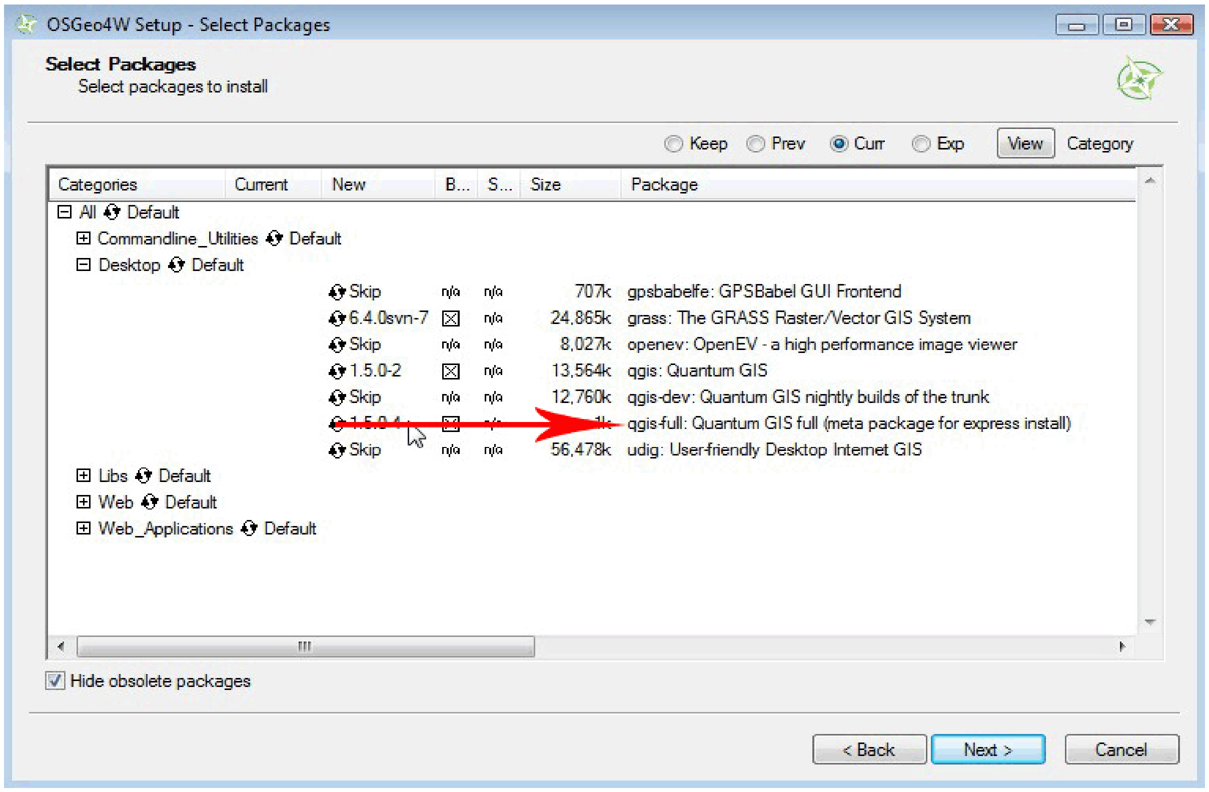
Quantum GIS, or QGIS, (http://www.qgis.org/) is another project of the Open Source Geospatial Foundation (OSGeo), which is a fully featured desktop GIS. It can operate on many different Operating Systems, and supports many vector, raster, database, and OGC formats.

One of the many capabilities provided by QGIS is the ability to export your QGIS styles to a MapServer mapfile.

Exercise 15: Add a WMS Layer in QGIS and Export to MapServer

Step1: Install QGIS Through OSGeo4W

In the Advanced interface for OSGeo4W, in the "Desktop" section, install the "qgis-full" package.


OSGeo4W Setup - Select Packages

Select Packages
Select packages to install

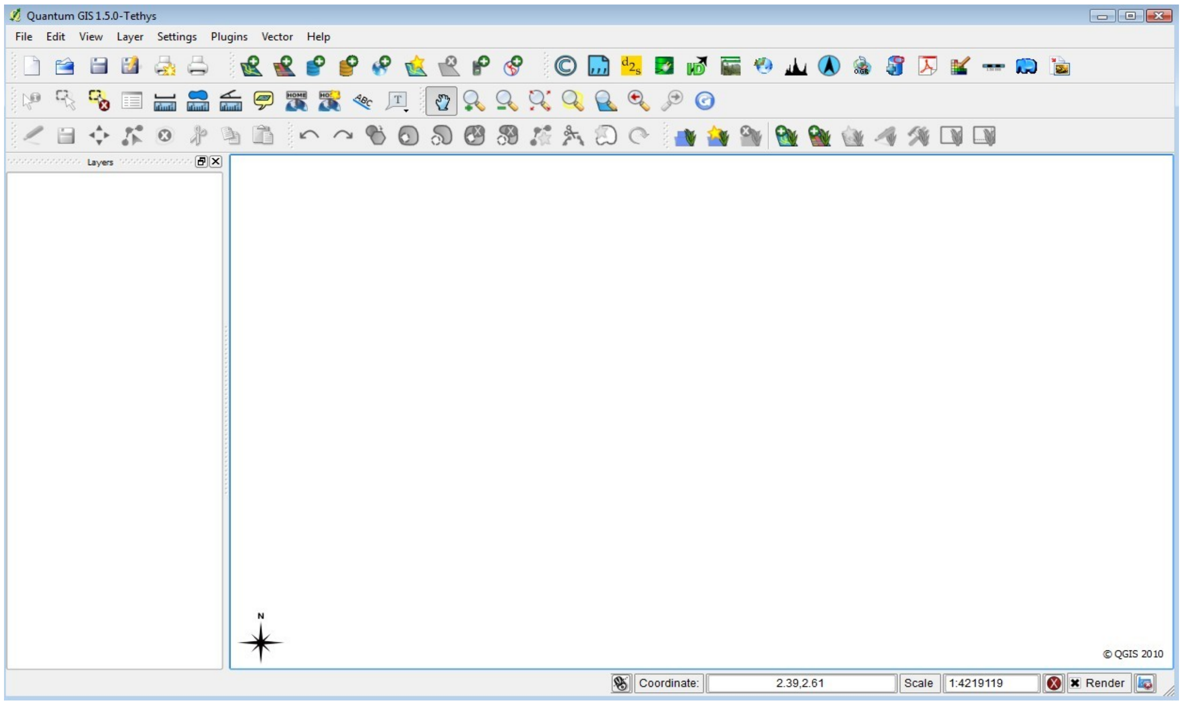
○ Keep ○ Prev ◉ Curr ○ Exp View Category

Categories	Current	New	B...	S...	Size	Package
⊟ All Default						
⊞ Commandline_Utilities Default						
⊟ Desktop Default						
		Skip	n/a	n/a	707k	gpsbabelfe: GPSBabel GUI Frontend
		6.4.0svn-7	☒	n/a	24,865k	grass: The GRASS Raster/Vector GIS System
		Skip	n/a	n/a	8,027k	openev: OpenEV - a high performance image viewer
		1.5.0-2	☒	n/a	13,564k	qgis: Quantum GIS
		Skip	n/a	n/a	12,760k	qgis-dev: Quantum GIS nightly builds of the trunk
		1.5.0-4	☒	n/a	1k	qgis-full: Quantum GIS full (meta package for express install)
		Skip	n/a	n/a	56,478k	udig: User-friendly Desktop Internet GIS
⊞ Libs Default						
⊞ Web Default						
⊞ Web_Applications Default						

☑ Hide obsolete packages

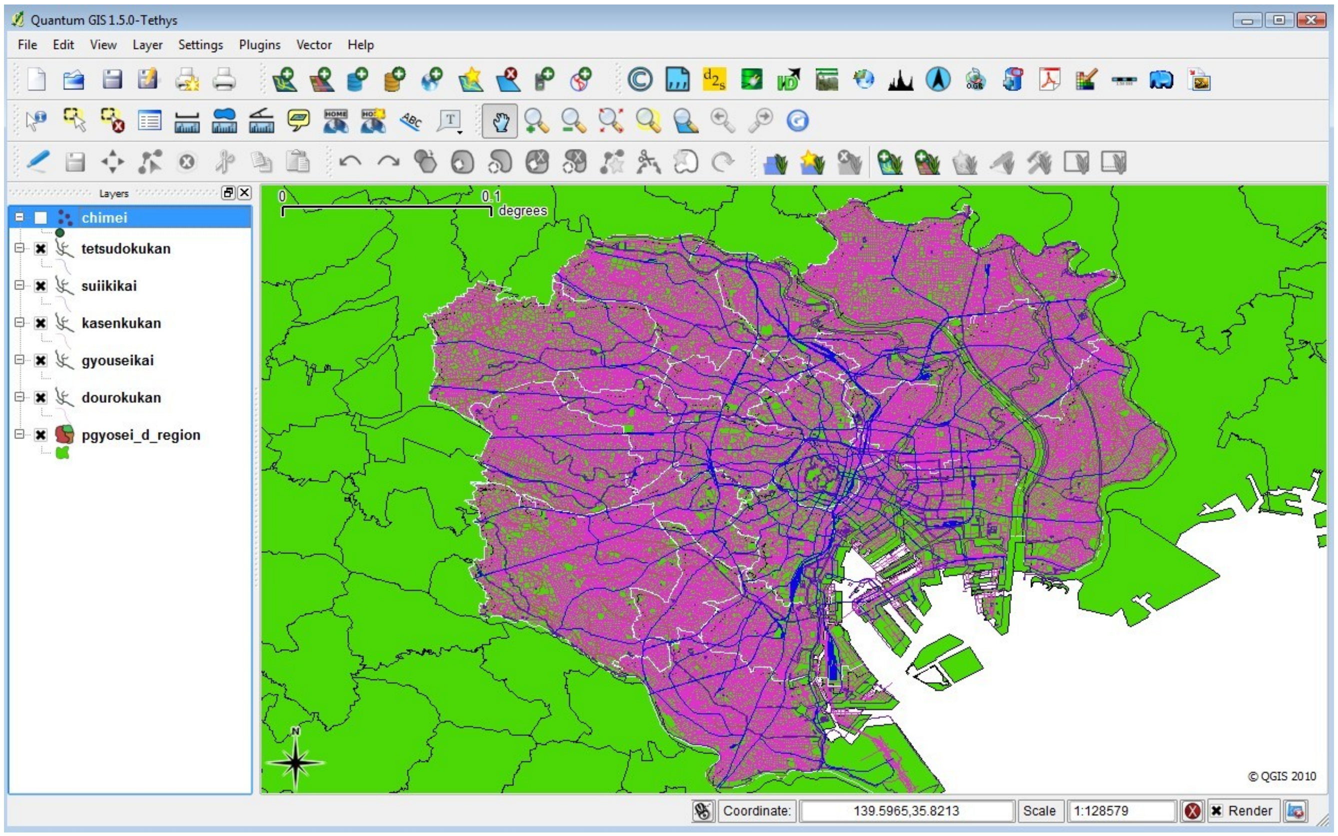
Step3: Start QGIS

Goto Start Menu/Programs/OSGeo4W and select "Quantum GIS".



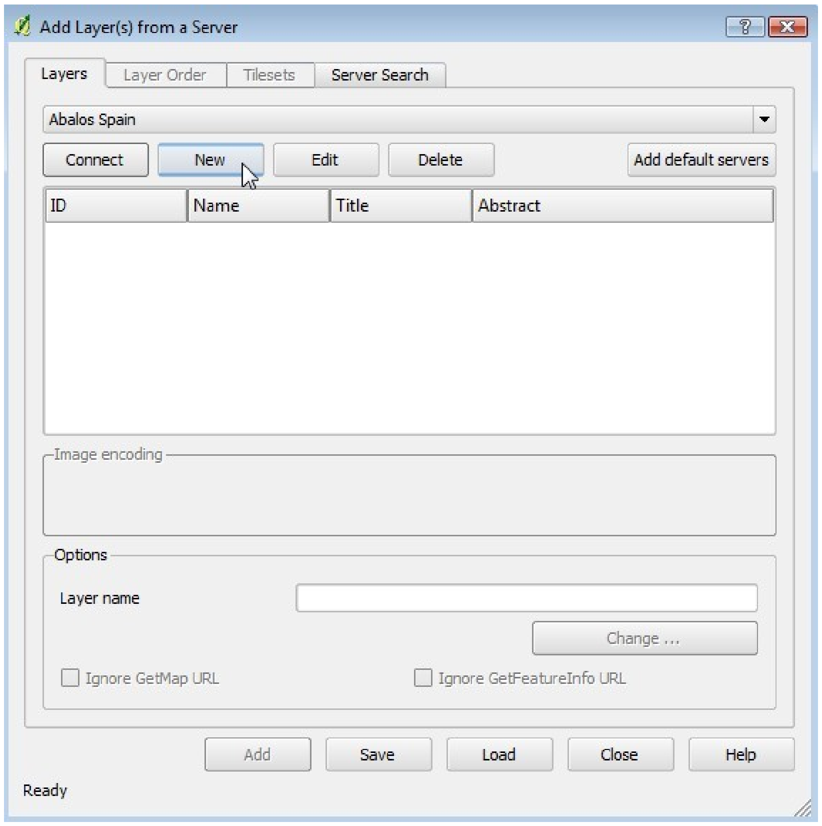
Step4: Add all of the Exercise Shapefiles

- in the Layer menu, select "Add vector layer"
- browse to the location of the files (OSGeo4W\apps\mapserver-tutorial\data\)
- select all shapefiles
- experiment with styling by clicking on the layer name in the legend

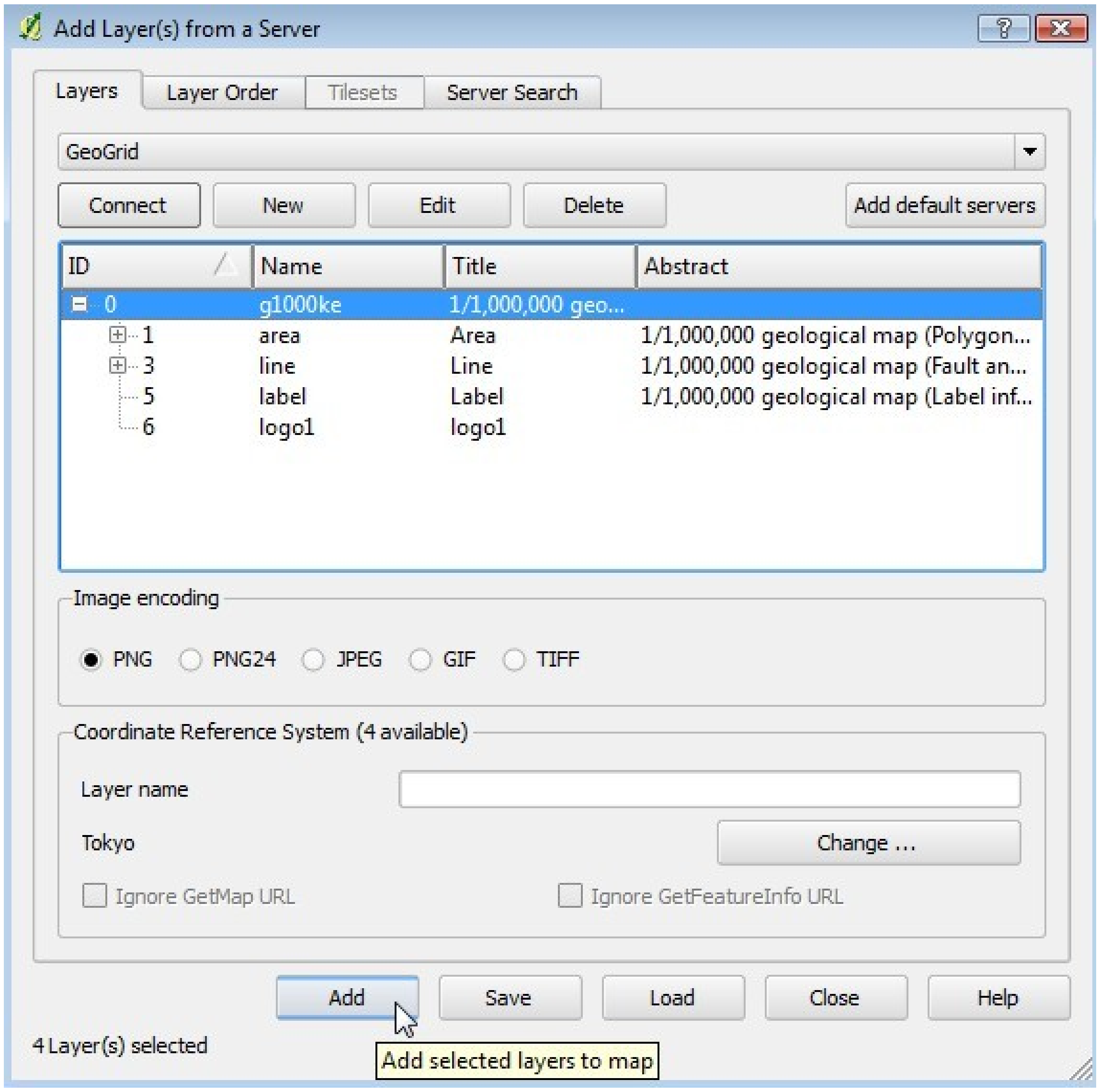


Step5: Add the GeoGrid WMS Layer

- in the Layer menu, select “Add WMS Layer”
- You should see the “Add Layer(s) from a Server” window
- Select “New”



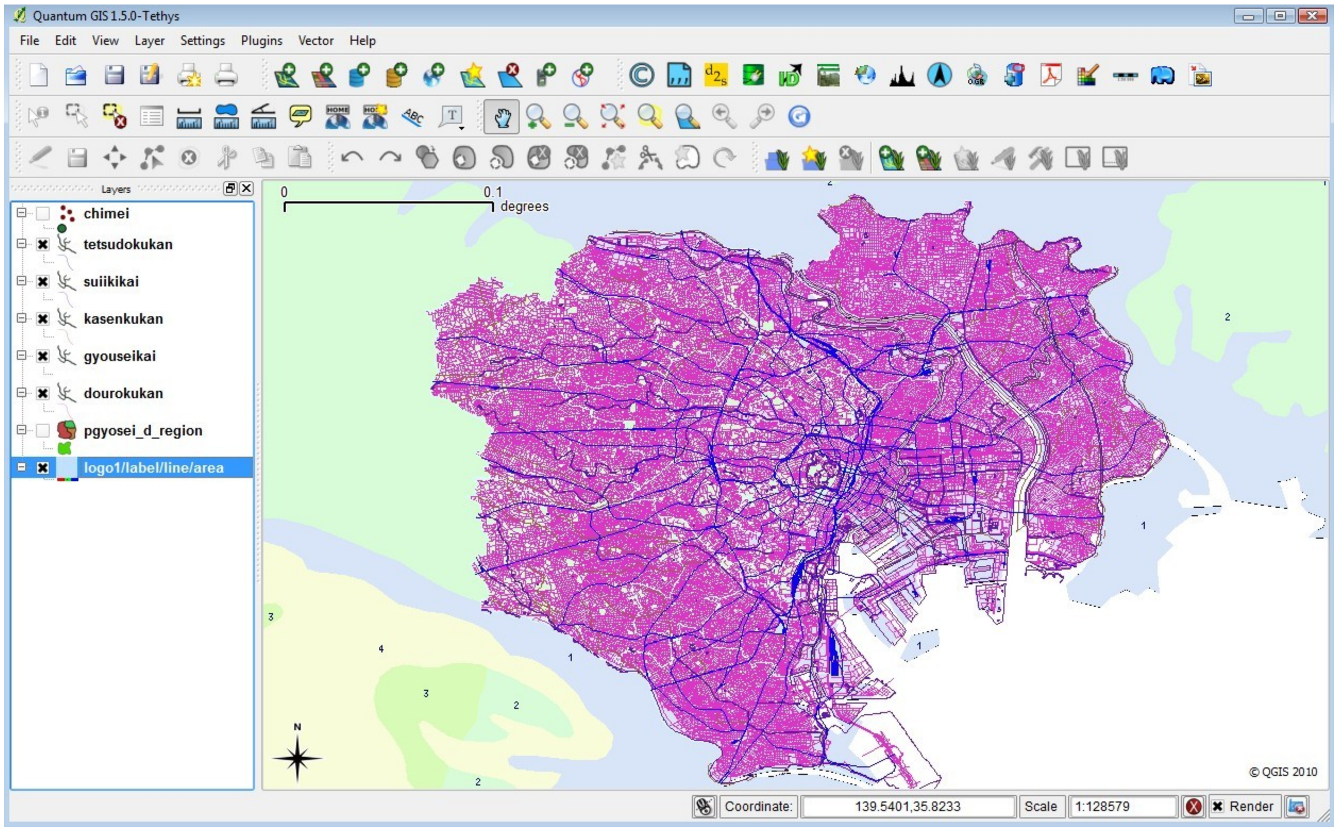
- In the "Create a new WMS connection" window enter "GeoGrid" for the name, and for the URL: http://geodata1.geogrid.org/mapserv/g1000k/g1000ke?
- From the dropdown in the "Add Layer(s) from a Server" window select the "GeoGrid" one
- Click "Connect" (the GetCapabilities will be parsed and the layer names will be read)
- Highlight the first layer, named "g1000ke" (see the following image) and click "Add"



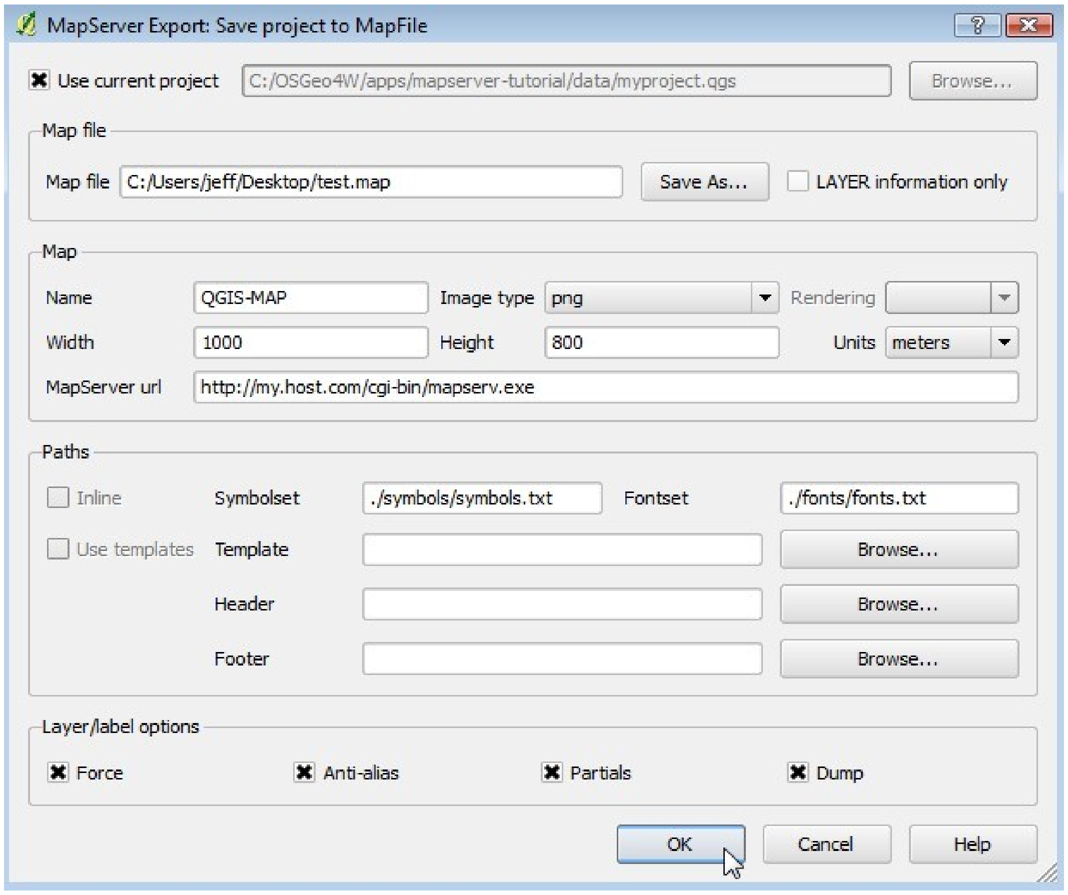
ID	Name	Title	Abstract
0	g1000ke	1/1,000,000 geo...	
1	area	Area	1/1,000,000 geological map (Polygon...
3	line	Line	1/1,000,000 geological map (Fault an...
5	label	Label	1/1,000,000 geological map (Label inf...
6	logo1	logo1	

Image encoding: [✓ PNG] PNG24 JPEG GIF TIFF

- close the “Add Layer(s) from a Server” window
- the GeoGrid layer should be overlayed on your map image
- reorder the layers in the legend so that the GeoGrid layer is at the bottom of the image, and turn off the 'pgyosei_d_region' layer



- save your project (File/Save Project)
- goto Plugins/MapServer Export
- the MapServer Export window is now displayed (see the following image). Configure the options you wish for your mapfile, and press <OK>



[✓] Use current project: C:/OSGeo4W/apps/mapserver-tutorial/data/myproject.qgs — Browse...

Map file

Map file: C:/Users/jeff/Desktop/test.map — Save As... — [ ] LAYER information only

Map

Name	QGIS-MAP	Image type	png	Rendering	
Width	1000	Height	800	Units	meters
MapServer url	http://my.host.com/cgi-bin/mapserv.exe				

Paths

[ ] Inline — Symbolset: ./symbols/symbols.txt — Fontset: ./fonts/fonts.txt

[ ] Use templates — Template: — Browse...

Header: — Browse...

Footer: — Browse...

Layer/label options

[✓] Force [✓] Anti-alias [✓] Partials [✓] Dump

- a mapfile should have been created with the styles that you chose, including a layer for the GeoGrid WMS server. It is recommended that you open the mapfile in a text editor and correct any mistakes. Test the mapfile with the shp2img commandline utility.

```
# Map file created from QGIS project file C:/OSGeo4W/apps/mapserver-
tutorial/data/myproject.qgs
# Edit this file to customize for your map interface
# (Created with PyQgis MapServer Export plugin)
MAP
  NAME "QGIS-MAP"
  # Map image size
  SIZE 1000 800
  UNITS meters

  EXTENT 139.485746 35.533769 139.994240 35.823828
  . . .
  . . .
  LAYER
    NAME 'logo1/label/line/area'
    TYPE RASTER
    DUMP true
    TEMPLATE fooOnlyForWMSGetFeatureInfo
  EXTENT 139.485746 35.533769 139.994240 35.823828
    CONNECTIONTYPE WMS
    CONNECTION 'http://geodata1.geogrid.org/mapserv/g1000k/g1000ke?'
    METADATA
      'ows_name' 'logo1,label,line,area'
      'wms_server_version' '1.1.1'
      'ows_srs' 'EPSG:4301'
      'wms_format' 'image/png'
      'wms_style' ',,,'
    END
    METADATA
      'ows_title' 'logo1/label/line/area'
    END
    STATUS OFF
    TRANSPARENCY 100
    PROJECTION
    'proj=longlat'
    'ellps=WGS84'
    'datum=WGS84'
    'no_defs'
    END
  END
```

Using WFS Services with MapServer

About WFS

Another popular standard introduced by the Open Geospatial Consortium is the Web Feature Service (WFS) specification, which shares features and attributes of vector data across remote servers in the form of GML (Geography Markup Language).

MapServer has always been one of the early adopters of OGC standards. MapServer's OGC standard support is well documented at: http://www.mapserver.org/ogc/ This section will discuss MapServer's WFS support. MapServer can act both as a WFS server (serving vector data in the form of GML to other servers), or as a WFS client (request and use vector data from other servers). Some relevant documents from MapServer are:

- MapServer WFS Server Howto: http://www.mapserver.org/ogc/wfs_server.html
- MapServer WFS Client Howto: http://www.mapserver.org/ogc/wfs_client.html

Exercise 16: Serving WFS Layers Through MapServer (WFS Server)

In this exercise we will add a few mandatory parameters to our mapfile so that it can be served through the WFS standard.

Step1: Configure Your Mapfile

A portion of the mapfile used in this exercise follows:

```
MAP
  NAME         "mapserver_tutorial_wfs_server"
  IMAGETYPE    PNG
  EXTENT       139.72520 35.67139 139.78845 35.70731
  SIZE         550 450

  IMAGECOLOR   255 255 255
  SHAPEPATH    "../data"

  FONTSET      ../fonts/fonts.txt
```

```
UNITS dd

WEB
  TEMPLATE  'example_template.html'
  IMAGEPATH "@osgeo4w@/tmp/ms_tmp/"
  IMAGEURL "/ms_tmp/"
  MINSCALEDENOM  1000
  MAXSCALEDENOM  70000
  METADATA
    "wfs_title"          "MapServer Tutorial WFS Server"
    "wfs_onlineresource" "http://127.0.0.1/cgi-bin/mapserv.exe?
MAP=@osgeo4w@/apps/mapserver-tutorial/example15.map"
    "wfs_srs"            "EPSG:4301 EPSG:4269 EPSG:4326"
    "wfs_abstract"      "This demonstration server shows how to setup a
MapServer .map file to serve data through the WFS standard."
  END
END

PROJECTION
  "init=epsg:4301"
END

LAYER
  NAME "roads"
  METADATA
  "wfs_title"               "Roads"
  "wfs_srs"                "EPSG:4301"
  "gml_include_items" "NAMAE,SYUBETSU"
  "gml_featureid"       "ID"
  END
  DATA dourokukan
  STATUS ON
  TYPE LINE
  MAXSCALEDENOM 50000
  CLASS
    NAME "roads"
```

```
      STYLE
        COLOR 187 187 127
      END
    END
    PROJECTION
      "init=epsg:4301"
    END
    DUMP TRUE
  END
  ...
END
```

The highlighed text shows the required parameters to serve data through the WFS standard with MapServer, and they are described next:

WFS_TITLE

This metadata must be set in the WEB object and in every layer that you want served through WFS. Use it to describe your data. All possible WMS metadata items are documented at: http://www.mapserver.org/ogc/wfs_server.html#reference-section

WFS_ONLINERESOURCE

This metadata item must be set in the WEB object; it is the URL to your mapserv.exe executable, and in our example includes the full path to our mapfile. More information about this metadata item can be read at: http://www.mapserver.org/ogc/wms_server.html#more-about-the-online-resource-url

GML_INCLUDE_ITEMS

This metadata item is recommended for each LAYER; it is the list of attributes (fields) that you would like to publish for the layer. If you would like all of the attributes to be shared, you can specify “all”.

GML_FEATUREID

This metadata item must be set for each LAYER; it is the field name to be used as the ID of the feature in the output GML.

PROJECTION

In order to serve data through any OGC standard, you must specify the projection that the data is in.

Each layer must have a PROJECTION object, as well as a PROJECTION set at the MAP object level. Please read the MapServer documentation on the PROJECTION object: http://www.mapserver.org/mapfile/projection.html

Step2: Validate Your Server's Capabilities

Now that we have configured our mapfile for serving WFS, we must check that it is configured properly for the standard. We do this by using the onlineresource URL in a browser, and add the following to it: “&SERVICE=WFS&VERSION=1.0.0&REQUEST=GetCapabilities “. In our example, the full request is:

```
http://127.0.0.1/cgi-bin/mapserv.exe?MAP=C:/OSGeo4W/apps/mapserver-
tutorial/example16/example.map&SERVICE=wfs&VERSION=1.0.0&REQUEST=GetCapabilities
```

Generally that request is referred to as the GetCapabilities request. If you execute it in a browser, you will be asked to save a file: rename this file so that it has a .xml extension (such as 'mapserv.xml'). Then open this file in a text editor. You should see a response from your WFS server, describing the layers available.

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<WFS_Capabilities version="1.0.0" updateSequence="0"
xmlns="http://www.opengis.net/wfs" xmlns:ogc="http://www.opengis.net/ogc"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs
http://schemas.opengis.net/wfs/1.0.0/WFS-capabilities.xsd">


<!-- MapServer version 5.6.5 OUTPUT=GIF OUTPUT=PNG OUTPUT=JPEG OUTPUT=WBMP
OUTPUT=SWF OUTPUT=SVG SUPPORTS=PROJ SUPPORTS=AGG SUPPORTS=FREETYPE SUPPORTS=ICONV
SUPPORTS=FRIBIDI SUPPORTS=WMS_SERVER SUPPORTS=WMS_CLIENT SUPPORTS=WFS_SERVER
SUPPORTS=WFS_CLIENT SUPPORTS=WCS_SERVER SUPPORTS=SOS_SERVER SUPPORTS=THREADS
SUPPORTS=RGBA_PNG INPUT=JPEG INPUT=POSTGIS INPUT=OGR INPUT=GDAL INPUT=SHAPEFILE -->


<Service>

  <Name>MapServer WFS</Name>

  <Title>MapServer Tutorial WFS Server</Title>

  <Abstract>This demonstration server shows how to setup a MapServer .map file to
serve data through the WFS standard.</Abstract>

  <OnlineResource>http://127.0.0.1/cgi-bin/mapserv.exe?
MAP=@osgeo4w@/apps/mapserver-tutorial/example15.map&amp;</OnlineResource>

</Service>
```

```
<Capability>
  <Request>
    <GetCapabilities>
    ...
<FeatureTypeList>
  <Operations>
    <Query/>
  </Operations>
    <FeatureType>
        <Name>roads</Name>
        <Title>Roads</Title>
        <SRS>EPSG:4301</SRS>
        <LatLongBoundingBox minx="139.563" miny="35.5247" maxx="139.918"
maxy="35.8176"/>
    </FeatureType>
    ...
</FeatureTypeList>

<ogc:Filter_Capabilities>
  <ogc:Spatial_Capabilities>
    <ogc:Spatial_Operators>
      <ogc:BBOX/>
    </ogc:Spatial_Operators>
  </ogc:Spatial_Capabilities>
  <ogc:Scalar_Capabilities>
    <ogc:Logical_Operators/>
    <ogc:Comparison_Operators>
      <ogc:Simple_Comparisons/>
      <ogc:Like/>
      <ogc:Between/>
    </ogc:Comparison_Operators>
  </ogc:Scalar_Capabilities>
</ogc:Filter_Capabilities>
</WFS_Capabilities>
```

If you are going to add a WFS layer from a remote server, it is recommended that you look at the server's GetCapabilities document. In our example above, you can see that our abstract and titles for all of the layers are displayed in the GetCapabilities document, as well as other important information such as what image formats the data is available in, and what projections (or "SRS") we can request the data in. If you have a problem with your WMS service, a "*WARNING*" tag will be included in your GetCapabilities document.

Step3: Request Features from your Server

Now that you have verified that the capabilities of your WFS server are valid, you can now request features from your WFS server. The WFS specification uses a *GetFeature* request to request GML features from a remote WFS server.

In our example, our GetFeature request looks like this:

```
http://127.0.0.1/cgi-bin/mapserv.exe?MAP=C:\OSGeo4W/apps/mapserver-
tutorial/example16/example.map&SERVICE=wfs&VERSION=1.0.0&REQUEST=GetFeature&TYPENAM
E=roads&SRS=EPSG:4301&BBOX=139.72520,35.67139,139.78845,35.70731&FORMAT=image/png&W
IDTH=550&HEIGHT=450
```

In our example, the GetFeature request will return GML such as the following:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<wfs:FeatureCollection xmlns:ms="http://mapserver.gis.umn.edu/mapserver"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:gml="http://www.opengis.net/gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.opengis.net/wfs
http://schemas.opengis.net/wfs/1.0.0/WFS-basic.xsd
http://mapserver.gis.umn.edu/mapserver http://127.0.0.1/cgi-bin/mapserv.exe?
MAP=@osgeo4w@/apps/mapserver-
tutorial/example15.map&amp;SERVICE=WFS&amp;VERSION=1.0.0&amp;REQUEST=DescribeFeatur
eType&amp;TYPENAME=roads&amp;OUTPUTFORMAT=XMLSCHEMA">
      <gml:boundedBy>
            <gml:Box srsName="EPSG:4301">
                  <gml:coordinates>139.722932,35.669905
139.789447,35.711632</gml:coordinates>
            </gml:Box>
      </gml:boundedBy>
    <gml:featureMember>
      <ms:roads fid="roads.DK13101001306">
```

```
        <gml:boundedBy>
            <gml:Box srsName="EPSG:4301">
                  <gml:coordinates>139.756261,35.674216 
139.756532,35.674510</gml:coordinates>
            </gml:Box>
        </gml:boundedBy>
        <ms:msGeometry>
        <gml:LineString srsName="EPSG:4301">
          <gml:coordinates>139.756532,35.674510 139.756404,35.674357 
139.756261,35.674216 </gml:coordinates>
        </gml:LineString>
        </ms:msGeometry>
        <ms:NAMAE/>
        <ms:SYUBETSU>&#8217;· #8240;&#8364;&#732;H</ms:SYUBETSU>
      </ms:roads>
    </gml:featureMember>
    <gml:featureMember>
      <ms:roads fid="roads.DK13101001305">
        <gml:boundedBy>
            <gml:Box srsName="EPSG:4301">
                  <gml:coordinates>139.755285,35.672010 
139.756261,35.674216</gml:coordinates>
            </gml:Box>
        </gml:boundedBy>
        <ms:msGeometry>
        <gml:LineString srsName="EPSG:4301">
          <gml:coordinates>139.756261,35.674216 139.756036,35.673507 
139.755856,35.673029 139.755285,35.672010 </gml:coordinates>
        </gml:LineString>
        </ms:msGeometry>
        <ms:NAMAE/>
        <ms:SYUBETSU>&#8217;· #8240;&#8364;&#732;H</ms:SYUBETSU>
      </ms:roads>
    </gml:featureMember>
</wfs:FeatureCollection>
```

Exercise 17: Consuming Remote WFS Layers in MapServer (WFS Client)

In this exercise we will add a layer from a remote WMS service to our mapfile.

Step1: Choose a remote WFS Service

Locating WFS services is not easy, as there is no official list of services (at the time of writing this document). For your benefit here are some possible WFS services:

Description	More Information	Onlineresource URL
Autodesk MapGuide WFS service	Demo WFS service hosted in by Japan Autodesk partner Applied Technology Group	http://www.autodesk-mapguide.net/mapguide2011/mapagent/mapagent.fcgi?
KSJ WFS	Simplified prefectures in Japan (polygon), Stations (line, not point). Data sources: Digital National Land Information, by National and Regional Planning Bureau, Ministry of Land, Infrastructure, Transport and Tourism.	http://www.finds.jp/ws/ksjwfs.cgi?
Placenames WFS	Prefecture names (point), Municipality (city, town or village) names (point). Data source: Digital National Land Information	http://www.finds.jp/ws/pnwfs.cgi?
MapServer Demo WFS service	MapServer WFS service, used in MapServer documentation	http://demo.mapserver.org/cgi-bin/wfs?

Step2: Review the WFS Server's GetCapabilities

In this example we chose the MapServer Demo WFS service (http://demo.mapserver.org/cgi-bin/wfs?). To look at their service's capabilities we would execute a GetCapabilities request, such as:

http://demo.mapserver.org/cgi-bin/wfs?
SERVICE=WFS&VERSION=1.0.0&REQUEST=GetCapabilities

A portion of their GetCapabilities response follows:

```
<FeatureTypeList>
  <Operations>
    <Query/>
  </Operations>
    <FeatureType>
        <Name>continents</Name>
        <Title>World continents</Title>
        <SRS>EPSG:4326</SRS>
        <LatLongBoundingBox minx="-180" miny="-90" maxx="180" maxy="83.6274" />
    </FeatureType>

    <FeatureType>
        <Name>cities</Name>
        <Title>World cities</Title>
        <SRS>EPSG:4326</SRS>
        <LatLongBoundingBox minx="-178.167" miny="-54.8" maxx="179.383"
maxy="78.9333" />
    </FeatureType>
</FeatureTypeList>
```

From the GetCapabilities response, we can see that there is a "continents" layer available, and that server is providing the data in a different projection than our local dataset (EPSG:4326).

Step3: Add a WFS Layer to our Mapfile

To add a WFS layer to our mapfile, you can follow the instructions in the MapServer WFS Client document: http://www.mapserver.org/ogc/wfs_client.html

Our example WFS layer follows:

```
  LAYER
  NAME "continents"
  TYPE POLYGON
  STATUS ON
  CONNECTION "http://demo.mapserver.org/cgi-bin/wfs?"
  CONNECTIONTYPE WFS
  METADATA
```

```
    "wfs_typename"            "continents"
    "wfs_version"             "1.0.0"
    "wfs_maxfeatures"         "10"
  END
  PROJECTION
    "init=epsg:4326"
  END
  CLASS
    NAME "Continents"
    STYLE
      COLOR 255 128 128
      OUTLINECOLOR 96 96 96
    END
  END
END # Layer
```

Here is a description of the important parameters of this layer:

TYPE

Since we are requesting a WFS layer, and a GML file should be returned locally, we can set the type to be a local POLYGON.

CONNECTION

This is the onlineresource URL for the remote WFS server.

CONNECTIONTYPE

Must be set to 'WFS'.

WFS_TYPENAME

This is the name of the remote WFS layer, as specified in the GetCapabilities response.

WFS_VERSION

This is the WFS version as obtained through the GetCapabilities response.

WFS_MAXFEATURES

This optional parameter will set the maximum number of features to be returned for this layer.

PROJECTION

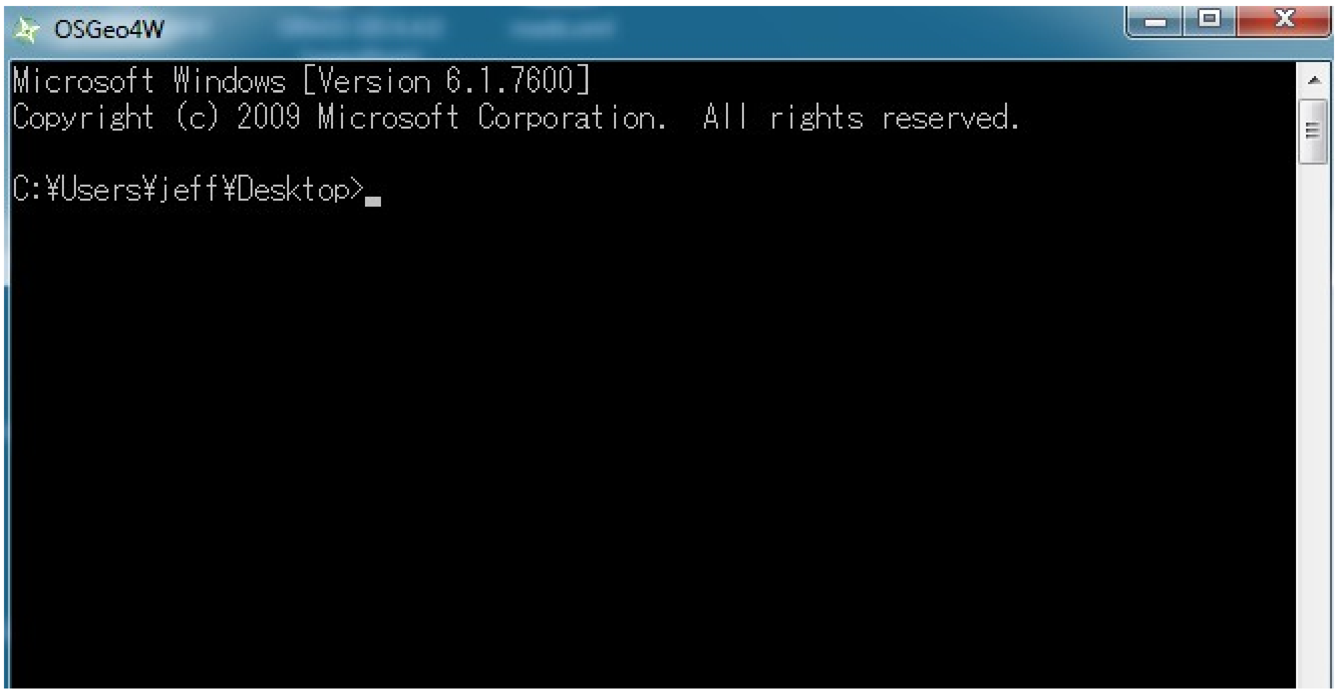
The layer must contain a projection object, which in this case is used to let MapServer know that the requested projection is EPSG:4326.

Step4: Generate a Map

Finally we can try to view our local layers with this remote WFS layer. The exercise HTML page has a working link to generate a map image, but you can also use MapServer's shp2img.exe commandline utility to test your mapfile. More information about shp2img can be found at: http://www.mapserver.org/utilities/shp2img.html

Execute the following steps:

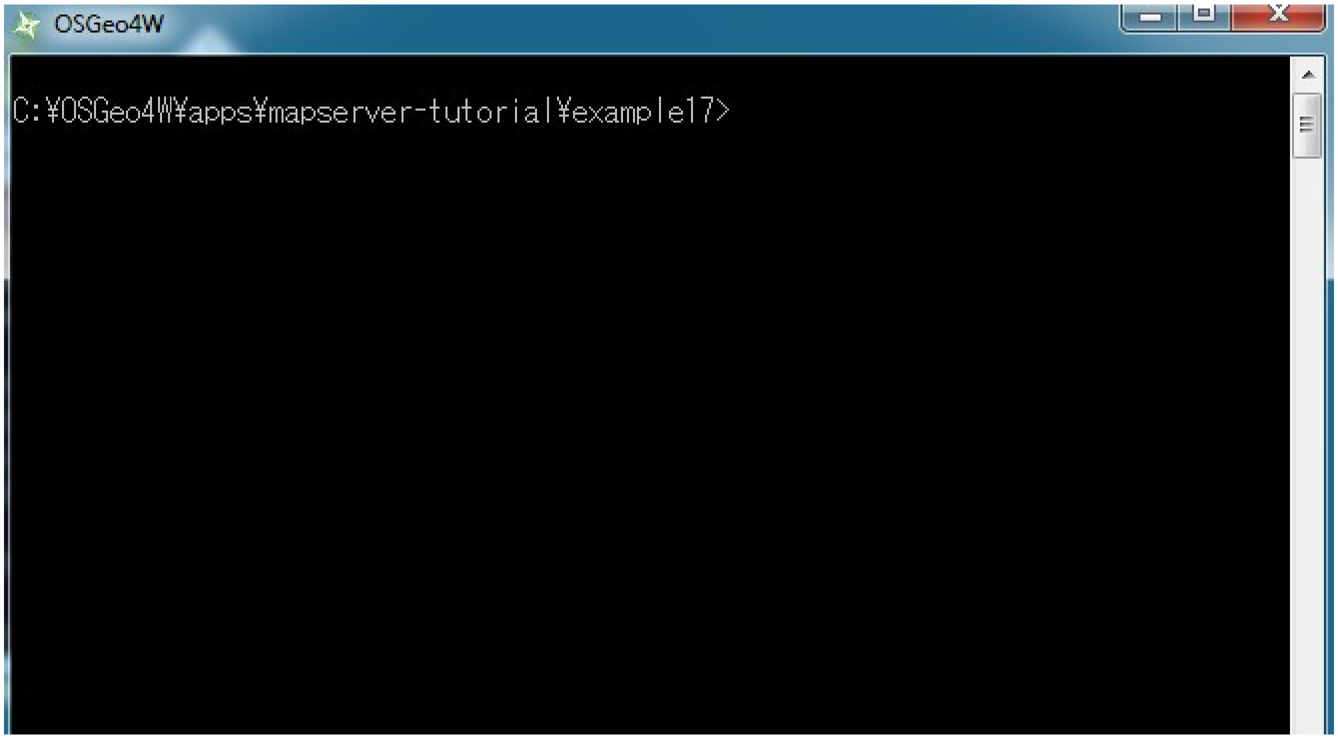
1) Click on the "OSGeo4W" shortcut on your desktop. A commandline shell window should open.



```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:¥Users¥jeff¥Desktop>
```

2) CD (change directory) to the example17 directory.

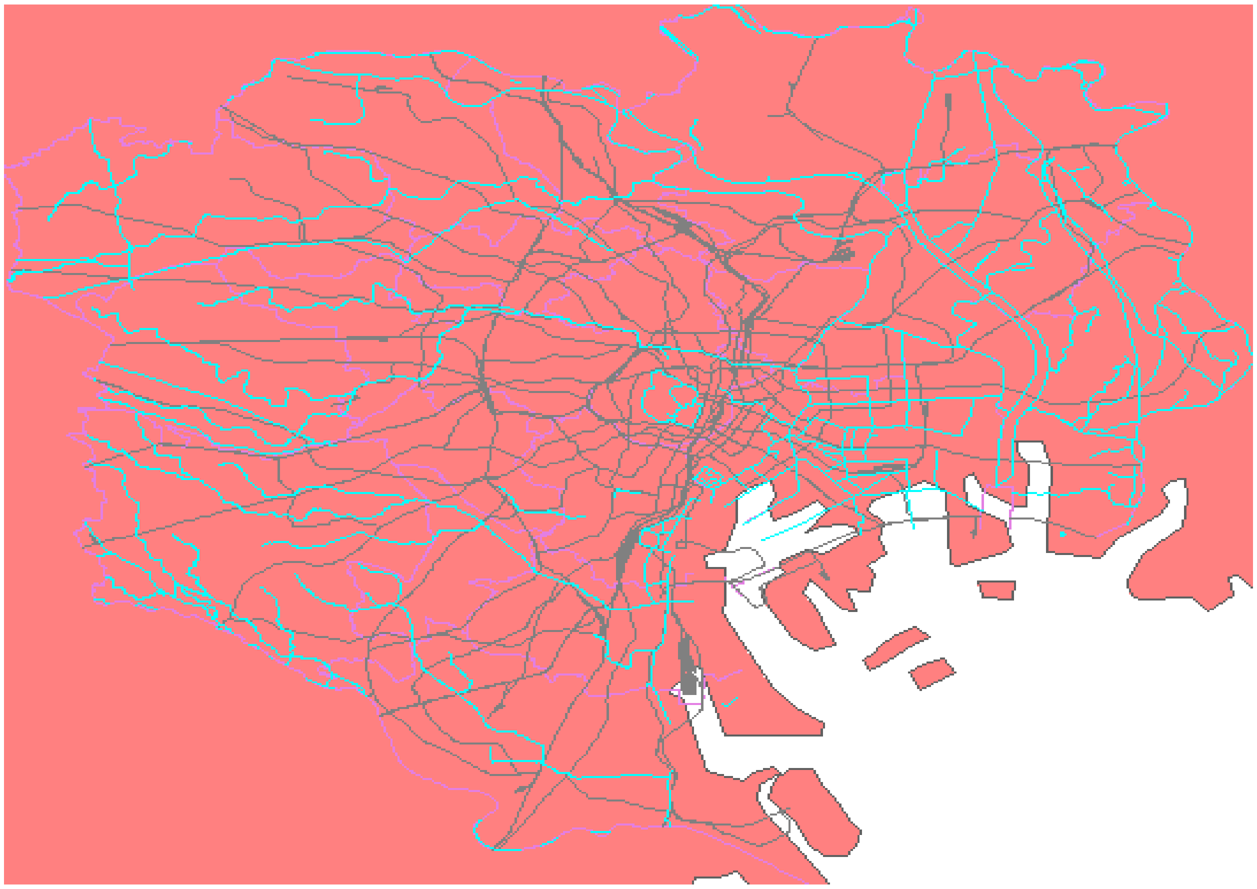


```
C:¥OSGeo4W¥apps¥mapserver-tutorial¥example17>
```

3) Execute: shp2img -m example.map -o test.png -map_debug 3
   (this should generate a 'test.png' map image, or display an error)

4) Drag 'test.png' into your Web browser.

   (if you have a blank image, open example17/example.map in a text editor and replace all 'STATUS OFF' values with 'STATUS ON' and retry shp2img)



You should now see a map image of our local layers overlayed on top of the remote WFS server (continents) layer.

Using WCS Services with MapServer

About WCS

Another standard introduced by the Open Geospatial Consortium is the Web Coverage Service (WCS) specification, which shares raster features through a multidimensional range of properties. The raster data is shared as coverages, allowing the client to execute complex queries against the data.

MapServer has always been one of the early adopters of OGC standards. MapServer's OGC standard support is well documented at: http://www.mapserver.org/ogc/ The following section will discuss MapServer's WCS support. MapServer can act as a WCS server (serving raster coverages to other servers), but MapServer cannot act as a WCS client. Some relevant documents from MapServer are:

- MapServer WCS Server Howto: http://www.mapserver.org/ogc/wcs_server.html
- MapServer WCS Use Cases: http://www.mapserver.org/ogc/wcs_format.html

Exercise 18: Serving WCS Layers Through MapServer

In this exercise we will add a few mandatory parameters to our mapfile so that it can be served through the WCS standard.

Step1: Configure Your Mapfile

The mapfile used in this exercise is:

```
MAP
NAME         "mapserver_tutorial_wcs_server"
IMAGETYPE    PNG
EXTENT       138.6464391 35.0053821 141.5283462 37.0799579
SIZE         550 450

IMAGECOLOR   255 255 255
SHAPEPATH    "../data"

FONTSET      ../fonts/fonts.txt
```

```
  UNITS dd

  WEB
    TEMPLATE  'example_template.html'
    IMAGEPATH "C:\OSGeo4W/tmp/ms_tmp/"
    IMAGEURL "/ms_tmp/"
    MINSCALEDENOM  1000
    MAXSCALEDENOM  70000
    METADATA
      "wcs_label"          "MapServer Tutorial WCS Server"
      "wcs_description"   "Demonstration WCS service serving a single Landsat image
of Tokyo (path:107 row:035)"
      "wcs_onlineresource" "http://127.0.0.1/cgi-bin/mapserv.exe?
MAP=C:\OSGeo4W/apps/mapserver-tutorial/example18.map"
    END
  END

  PROJECTION
    "init=epsg:4301"
  END

  LAYER
    NAME "landsat"
    METADATA
      "wcs_label"                 "Landsat"
      "wcs_rangeset_name"         "Range 1"
      "wcs_rangeset_label"        "Range 1 Label"
    END
    DATA "tokyo-landsat-p107-r035.tif"
    TYPE RASTER
    STATUS ON
    PROJECTION
      "init=epsg:4301"
    END
    DUMP TRUE
  END
END
```

The highlighed text shows the required parameters to serve data through the WCS standard with MapServer, and they are described next:

WCS_LABEL

This metadata must be set in the WEB object and in every layer that you want served through WCS. Use it to describe your data. All possible WMS metadata items are documented at: http://www.mapserver.org/ogc/wcs_server.html#reference-section

WCS_DESCRIPTION

This metadata must be set in the WEB object and can be used for a longer description of your WCS service.

WCS_ONLINERESOURCE

This metadata item must be set in the WEB object; it is the URL to your mapserv.exe executable, and in our example includes the full path to our mapfile. More information about this metadata item can be read at: http://www.mapserver.org/ogc/wms_server.html#more-about-the-online-resource-url

WCS_RANGESET_NAME

This metadata item must be set in each LAYER object, and can be used to describe the properties assigned to the coverage.

WCS_RANGESET_LABEL

This metadata item must be set in each LAYER object, and can be used to describe the properties assigned to the coverage.

PROJECTION

In order to serve data through any OGC standard, you must specify the projection that the data is in. Each layer must have a PROJECTION object, as well as a PROJECTION set at the MAP object level. Please read the MapServer documentation on the PROJECTION object: http://www.mapserver.org/mapfile/projection.html

DUMP TRUE

To serve the layer's data as a WCS coverage, you must set the DUMP TRUE parameter in your LAYER object.

Step2: Validate Your Server's Capabilities

Now that we have configured our mapfile for serving WCS, we must check that it is configured properly for the standard. We do this by using the onlineresource URL in a browser, and add the following to it: "&SERVICE=WCS&VERSION=1.0.0&REQUEST=GetCapabilities ". In our example, the full request is:

```
http://127.0.0.1/cgi-bin/mapserv.exe?MAP=C:/OSGeo4W/apps/mapserver-
tutorial/example18/example.map&SERVICE=wcs&VERSION=1.0.0&REQUEST=GetCapabilities
```

Generally that request is referred to as the GetCapabilities request. If you execute it in a browser, you will be asked to save a file: rename this file so that it has a .xml extension (such as 'mapserv.xml'). Then open this file in a text editor. You should see a response from your WCS server, describing the layers available, such as:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>

<WCS_Capabilities version="1.0.0" updateSequence="0"
xmlns="http://www.opengis.net/wcs" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:gml="http://www.opengis.net/gml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.opengis.net/wcs
http://schemas.opengis.net/wcs/1.0.0/wcsCapabilities.xsd">

<Service>

  <description>Demonstration WCS service serving a single Landsat image of Tokyo
(path:107 row:035)</description>

  <name>MapServer WCS</name>

  <label>MapServer Tutorial WCS Server</label>

  <fees>NONE</fees>

  <accessConstraints>

    NONE
  </accessConstraints>
</Service>
<Capability>
  <Request>

    <GetCapabilities>
      <DCPType>
        <HTTP>
          <Get><OnlineResource xlink:type="simple"
xlink:href="http://127.0.0.1/cgi-bin/mapserv.exe?MAP=C:\OSGeo4W/apps/mapserver-
tutorial/example18.map&amp;"/></Get>
        </HTTP>
      </DCPType>
      <DCPType>
        <HTTP>
          <Post><OnlineResource xlink:type="simple"
xlink:href="http://127.0.0.1/cgi-bin/mapserv.exe?MAP=C:\OSGeo4W/apps/mapserver-
tutorial/example18.map&amp;"/></Post>
        </HTTP>
      </DCPType>
    </GetCapabilities>
    <DescribeCoverage>
      <DCPType>
        <HTTP>
          <Get><OnlineResource xlink:type="simple"
xlink:href="http://127.0.0.1/cgi-bin/mapserv.exe?MAP=C:\OSGeo4W/apps/mapserver-
tutorial/example18.map&amp;"/></Get>
        </HTTP>
      </DCPType>
      <DCPType>
        <HTTP>
          <Post><OnlineResource xlink:type="simple"
xlink:href="http://127.0.0.1/cgi-bin/mapserv.exe?MAP=C:\OSGeo4W/apps/mapserver-
tutorial/example18.map&amp;"/></Post>
        </HTTP>
      </DCPType>
    </DescribeCoverage>
    <GetCoverage>
      <DCPType>
        <HTTP>
          <Get><OnlineResource xlink:type="simple" 
xlink:href="http://127.0.0.1/cgi-bin/mapserv.exe?MAP=C:\OSGeo4W/apps/mapserver-
tutorial/example18.map&amp;"/></Get>


        </HTTP>
      </DCPType>
      <DCPType>
        <HTTP>
          <Post><OnlineResource xlink:type="simple" 
xlink:href="http://127.0.0.1/cgi-bin/mapserv.exe?MAP=C:\OSGeo4W/apps/mapserver-
tutorial/example18.map&amp;"/></Post>
        </HTTP>
      </DCPType>
    </GetCoverage>
  </Request>
  <Exception>
    <Format>application/vnd.ogc.se_xml</Format>
  </Exception>
</Capability>
<ContentMetadata>
  <CoverageOfferingBrief>
  <name>landsat</name>
  <label>Landsat</label>
    <lonLatEnvelope srsName="urn:ogc:def:crs:OGC:1.3:CRS84">
      <gml:pos>138.646439071208 35.0053821042488</gml:pos>
      <gml:pos>141.528346228667 37.0799578740914</gml:pos>
    </lonLatEnvelope>
  </CoverageOfferingBrief>
</ContentMetadata>
</WCS_Capabilities>
```

If you are going to add a WCS layer from a remote server, it is recommended that you look at the server's GetCapabilities document.

Step3: Execute a DescribeCoverage

Now that we have verified the service with a GetCapabilities command, we can check a coverage offered by our WCS service. We do this by using the onlineresource URL in a browser, and add the following to it: "&SERVICE=WCS&VERSION=1.0.0&REQUEST=DescribeCoverage&COVERAGE=landsat ". In our example, the full request is:

```
http://127.0.0.1/cgi-bin/mapserv.exe?MAP=C:/OSGeo4W/apps/mapserver-tutorial/example18/example.map&SERVICE=wcs&VERSION=1.0.0&REQUEST=DescribeCoverage&COVERAGE=landsat
```

You should see a response from your WCS server, describing the layers available, such as:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<CoverageDescription version="1.0.0" updateSequence="0"
xmlns="http://www.opengis.net/wcs" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:gml="http://www.opengis.net/gml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.opengis.net/wcs
http://schemas.opengis.net/wcs/1.0.0/describeCoverage.xsd">
  <CoverageOffering>
  <name>landsat</name>
  <label>Landsat</label>
    <lonLatEnvelope srsName="urn:ogc:def:crs:OGC:1.3:CRS84">
      <gml:pos>138.646439071208 35.0053821042488</gml:pos>
      <gml:pos>141.528346228667 37.0799578740914</gml:pos>
    </lonLatEnvelope>
    <domainSet>
      <spatialDomain>
        <gml:Envelope srsName="EPSG:4326">
          <gml:pos>138.646439071208 35.0053821042488</gml:pos>
          <gml:pos>141.528346228667 37.0799578740914</gml:pos>
        </gml:Envelope>
        <gml:Envelope srsName="EPSG:4301">
          <gml:pos>138.646439071208 35.0053821042488</gml:pos>
          <gml:pos>141.528346228667 37.0799578740914</gml:pos>
        </gml:Envelope>
        <gml:RectifiedGrid dimension="2">
          <gml:limits>
```

```
            <gml:GridEnvelope>
              <gml:low>0 0</gml:low>
              <gml:high>9862 7099</gml:high>
            </gml:GridEnvelope>
          </gml:limits>

          <gml:axisName>x</gml:axisName>
          <gml:axisName>y</gml:axisName>
          <gml:origin>
            <gml:pos>138.646439071208 37.0799578740914</gml:pos>
          </gml:origin>
          <gml:offsetVector>0.000292193770400368 0</gml:offsetVector>
          <gml:offsetVector>0 -0.000292193770400368</gml:offsetVector>
        </gml:RectifiedGrid>
      </spatialDomain>
    </domainSet>
    <rangeSet>
      <RangeSet>
        <name>Range 1</name>
        <label>Range 1 Label</label>
      </RangeSet>
    </rangeSet>
    <supportedCRSs>
      <requestResponseCRSs>EPSG:4301</requestResponseCRSs>
      <nativeCRSs>EPSG:4301</nativeCRSs>
    </supportedCRSs>
    <supportedFormats>
      <formats>GTiff</formats>
    </supportedFormats>
    <supportedInterpolations default="nearest neighbor">
      <interpolationMethod>nearest neighbor</interpolationMethod>
      <interpolationMethod>bilinear</interpolationMethod>
    </supportedInterpolations>
  </CoverageOffering>
</CoverageDescription>
```

Step4: Try Your Service with a WCS Client

As MapServer cannot act as a WCS client, please check the list of compliant WCS products and choose your preferred client to test your service:

http://www.opengeospatial.org/resource/products/compliant